



Программный комплекс Систэм Платформ

SePlatform.Imitator 2.1

Руководство администратора

Редакция
3. Предварительная

Соответствует версии ПО
2.1.2



© ООО «СИСТЭМ СОФТ», 2022-2024. Все права защищены.

Авторские права на данный документ принадлежат ООО «СИСТЭМ СОФТ». Копирование, перепечатка и публикация любой части или всего документа не допускается без письменного разрешения правообладателя.

Содержание

1. Назначение и принцип работы	5
1.1. Воспроизведение истории	5
1.2. Перезапись исторических данных	5
1.3. Дополнение пропущенных исторических данных	6
2. Подготовка к работе	7
2.1. Установка SePlatform.Imitator	7
2.1.1. ОС Windows	7
2.1.2. ОС Linux	7
2.2. Запуск и останов SePlatform.Imitator	8
2.2.1. ОС Windows	8
2.2.2. ОС Linux	8
2.3. Удаление SePlatform.Imitator	8
2.3.1. ОС Windows	8
2.3.2. ОС Linux	8
3. Настройка SePlatform.Imitator в SePlatform.Development Studio	10
3.1. Добавление SePlatform.Imitator в проект	10
3.2. Настройки БД	12
3.2.1. Добавление БД в SePlatform.Historian	12
3.2.2. Сохранение истории в БД	13
3.3. Настройка источника данных	14
3.4. Настройка приложения SePlatform.Data Server и SePlatform.Imitator	16
3.5. Настройка логических типов	17
3.6. Настройки для режима воспроизведения истории	19
3.6.1. Настройка БД для воспроизведения истории	19
3.6.2. Настройка сигналов для воспроизведения истории	20
3.6.3. Требуемые модули SePlatform.Imitator	21
3.7. Настройки для режима перезаписи истории	21
3.7.1. Настройка БД для перезаписи истории	21
3.7.2. Настройка сигналов для перезаписи истории	22
3.7.3. Требуемые модули SePlatform.Imitator	24
3.8. Настройки для режима дополнения истории	24
3.8.1. Настройка БД для дополнения истории	24
3.8.2. Настройка сигналов для дополнения истории	25
3.8.3. Настройка передачи данных через файловый интерфейс	25
3.9. Применение конфигураций SePlatform.Data Server и SePlatform.Imitator	29
4. Настройка SePlatform.Imitator в Конфигураторе	30
4.1. Настройка конфигурации для режима воспроизведения истории	30
4.1.1. Настройка сигналов	30
4.1.2. Требуемые модули	30
4.1.3. Настройка сохранения данных в имитационную БД	31
4.2. Настройка конфигурации для режима перезаписи истории	31
4.2.1. Настройка сигналов	32
4.2.2. Требуемые модули	32
4.2.3. Настройка сохранения данных в БД	33
4.3. Настройка конфигурации для режима дополнения истории	33
4.3.1. Настройка сигналов	34
4.3.2. Требуемые модули	34
4.3.3. Настройка сохранения данных в БД	34
4.3.4. Настройка файлового интерфейса	35
4.4. Сохранение конфигурации SePlatform.Data Server	36

4.5. Загрузка конфигурации в SePlatform.Imitator	36
5. Служебные сигналы SePlatform.Imitator	37
6. Работа с SePlatform.Imitator	41
6.1. Воспроизведение истории	41
6.2. Перезапись исторических данных	44
6.2.1. Перезапись единственного значения	45
6.2.2. Перезапись массива данных	49
6.3. Дополнение пропущенных исторических данных	56
7. Приложения	57
Приложение А: Создание имитационной базы данных	57
Приложение В: Формат datetime_json	58
Приложение С: Формат trend_json	59

1. Назначение и принцип работы

SePlatform.Imitator - компонент Систэм Платформ, предназначенный для выполнения следующих функций:

- воспроизведения истории технологического процесса;
- перезаписи исторических данных;
- дополнения пропущенных исторических данных.

Общая схема системы с SePlatform.Imitator приведена ниже.



SePlatform.Data Server получает данные от нижестоящих систем и выполняет логическую обработку полученных данных. Затем SePlatform.Data Server записывает вычисленные значения в SePlatform.Historian. SePlatform.Imitator в зависимости от режима работы:

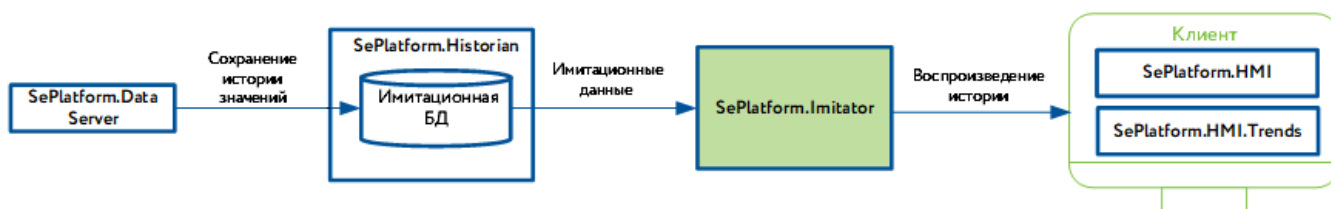
- воспроизводит историю изменения значений сигналов и передаёт клиенту;
- пересчитывает значения и перезаписывает скорректированные данные в SePlatform.Historian;
- записывает пропущенные данные в SePlatform.Historian.

Для выполнения каждой из перечисленных функций SePlatform.Imitator нужно запустить в соответствующем режиме работы.

1.1. Воспроизведение истории

SePlatform.Imitator позволяет загружать исторические данные из SePlatform.Historian за указанный интервал времени и проигрывать исторические данные в виде потока оперативных данных. Воспроизведение истории позволяет пользователю просмотреть историю хода технологического процесса на графиках или мнемосхемах.

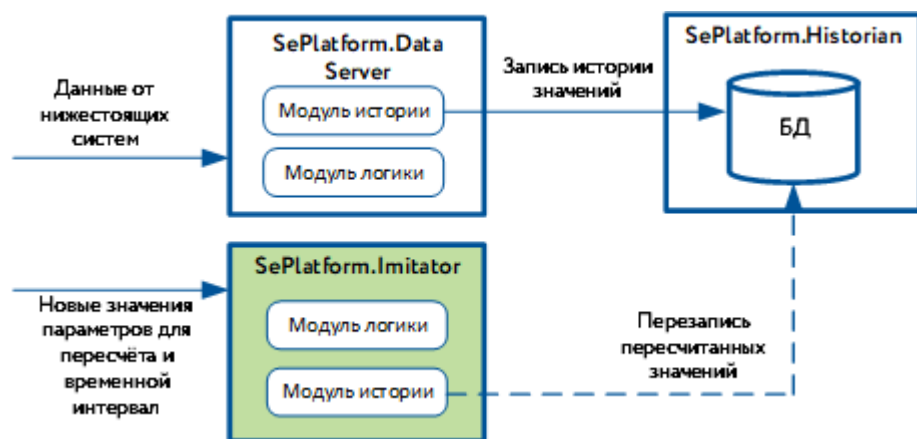
Для режима воспроизведения истории необходима отдельная база для имитационных данных.



1.2. Перезапись исторических данных

SePlatform.Imitator позволяет выполнять пересчёт значений на основе параметров, значения которых задаёт пользователь, и записывать пересчитанные значения в SePlatform.Historian за определенный промежуток

времени вместо уже имеющихся исторических данных с теми же метками времени.



SePlatform.Imitator может перезаписывать как единственное значение, так и массив данных.

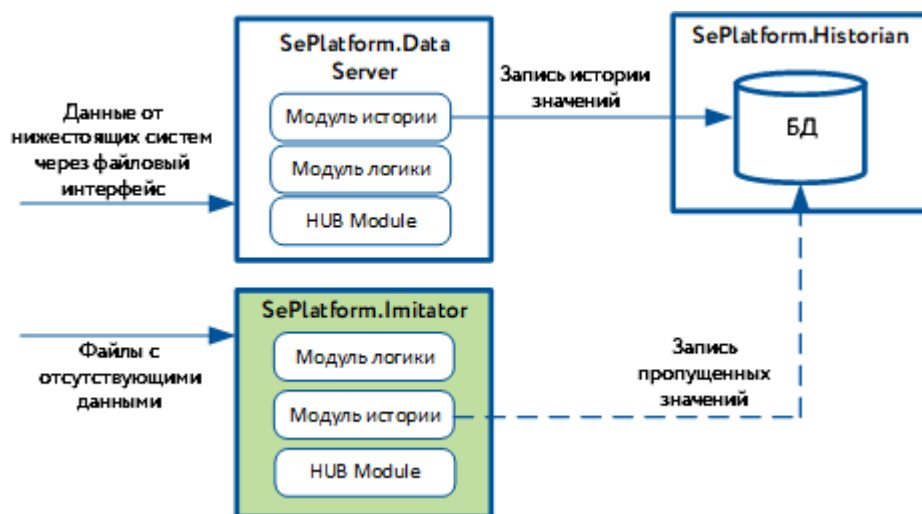
Перезапись единственного значения используется в случае, если требуется перезаписать значение тега, на основе которого будут пересчитаны зависимые параметры. При этом единственное заданное значение тега будут действовать на всём интервале имитации.

Перезапись массива данных используется в случае, если требуется перезаписать несколько значений тега, на основе которых будут пересчитаны зависимые параметры. При этом каждое из заданных значений будет действовать в соответствующем промежутке интервала имитации.

1.3. Дополнение пропущенных исторических данных

Если через файловый интерфейс SePlatform.Data Server не удаётся получить все файлы данных (например, из-за некорректной передачи файлов по сети), то в SePlatform.Historian в определенные промежутки времени исторические данные будут отсутствовать.

SePlatform.Imitator позволяет записывать отсутствующие данные в SePlatform.Historian. Для этого следует поместить непереданные файлы в заданную папку и запустить SePlatform.Imitator в режиме дополнения пропущенных данных. На основе данных, содержащихся в файлах, SePlatform.Imitator пересчитывает значения и записывает в соответствующие временные промежутки в SePlatform.Historian.



2. Подготовка к работе

SePlatform.Imitator функционирует в виде:

- службы `SePlatform.Imitator` в ОС Windows;
- сервиса `seplatform.imitator` в ОС Linux.

2.1. Установка SePlatform.Imitator

2.1.1. ОС Windows

Для установки SePlatform.Imitator запустите установочный файл `SePlatform.Imitator-x.x.x xxxxxx (x64).msi` и следуйте инструкциям мастера установки.

Установка выполняется в папку: `C:\Program Files\SePlatform\SePlatform.Imitator`.

2.1.2. ОС Linux



ОБРАТИТЕ ВНИМАНИЕ

Команда установки выполняется только от суперпользователя «root».

Имя устанавливаемого пакета: `seplatform.imitator-x.x.x+xx.xxxxx.deb` или `seplatform.imitator-x.x.x+xx.xxxxx.rpm` в зависимости от используемой ОС Linux. Находясь в папке с установочным пакетом, запустите установку.

Установка пакета *.rpm с помощью пакетного менеджера YUM:

```
yum install seplatform.imitator-x.x.x+xx.xxxxx.rpm
```

Установка пакета *.rpm с помощью пакетного менеджера RPM:

```
rpm -i seplatform.imitator-x.x.x+xx.xxxxx.rpm
```

Установка пакета *.deb с помощью пакетного менеджера apt:

```
apt-get install seplatform.imitator-x.x.x+xx.xxxxx.deb
```

Установка пакета *.deb с помощью пакетного менеджера dpkg:

```
sudo dpkg -i seplatform.imitator-x.x.x+xx.xxxxx.deb
```

SePlatform.Imitator устанавливается в директорию `/opt/SePlatform/SePlatform.Imitator`.

2.2. Запуск и останов SePlatform.Imitator

2.2.1. ОС Windows

Управление SePlatform.Imitator выполняется путем запуска/перезапуска/останова службы **SePlatform.Imitator** стандартными инструментами ОС Windows.

2.2.2. ОС Linux

Управление SePlatform.Imitator выполняется путем запуска/перезапуска/останова сервиса **seplatform.imitator** специализированными командами.



ОБРАТИТЕ ВНИМАНИЕ

Все команды выполняются только от суперпользователя «root».

Запуск:

```
systemctl start seplatform.imitator
```

Останов:

```
systemctl stop seplatform.imitator
```

Перезапуск:

```
systemctl restart seplatform.imitator
```

2.3. Удаление SePlatform.Imitator

2.3.1. ОС Windows

Удаление SePlatform.Imitator выполняется стандартными инструментами ОС Windows:

1. Запустите **Программы и компоненты**: Пуск → Панель управления → Программы и компоненты.
2. Из представленного списка установленных программ выберите SePlatform.Imitator и нажмите кнопку **Удалить**.

2.3.2. ОС Linux

Удаление SePlatform.Imitator выполняется штатным пакетным менеджером.



ОБРАТИТЕ ВНИМАНИЕ

Команда удаления выполняется только от суперпользователя «root».

Удаление пакета *.rpm с помощью пакетного менеджера YUM:


```
yum remove seplatform.imitator
```

Удаление пакета *.rpm с помощью пакетного менеджера RPM:

```
rpm -e seplatform.imitator
```

Удаление пакета *.deb с помощью пакетного менеджера apt:

```
apt-get remove seplatform.imitator
```

Удаление пакета *.deb с помощью пакетного менеджера dpkg:

```
dpkg -r seplatform.imitator
```

3. Настройка SePlatform.Imitator в SePlatform.Development Studio

Чтобы настроить SePlatform.Imitator:

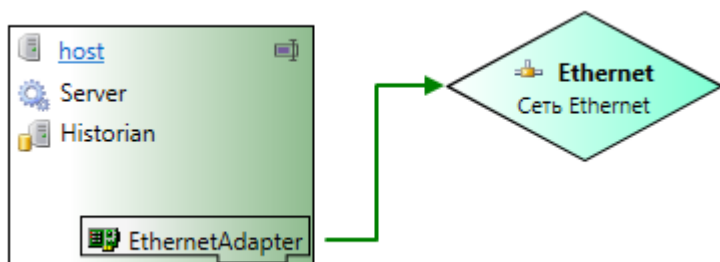
- в проект добавьте SePlatform.Imitator и выполните настройку SePlatform.Domain ([стр. 10](#));
- выполните настройки БД: настройте сохранение исторических данных SePlatform.Data Server в БД SePlatform.Historian ([стр. 12](#));
- настройте источник данных для SePlatform.Data Server ([стр. 14](#));
- настройте приложение для SePlatform.Data Server и SePlatform.Imitator ([стр. 16](#));
- выполните настройки, необходимые для требуемого режима работы SePlatform.Imitator;
- примените конфигурации SePlatform.Data Server и SePlatform.Imitator ([стр. 29](#)).



ОБРАТИТЕ ВНИМАНИЕ

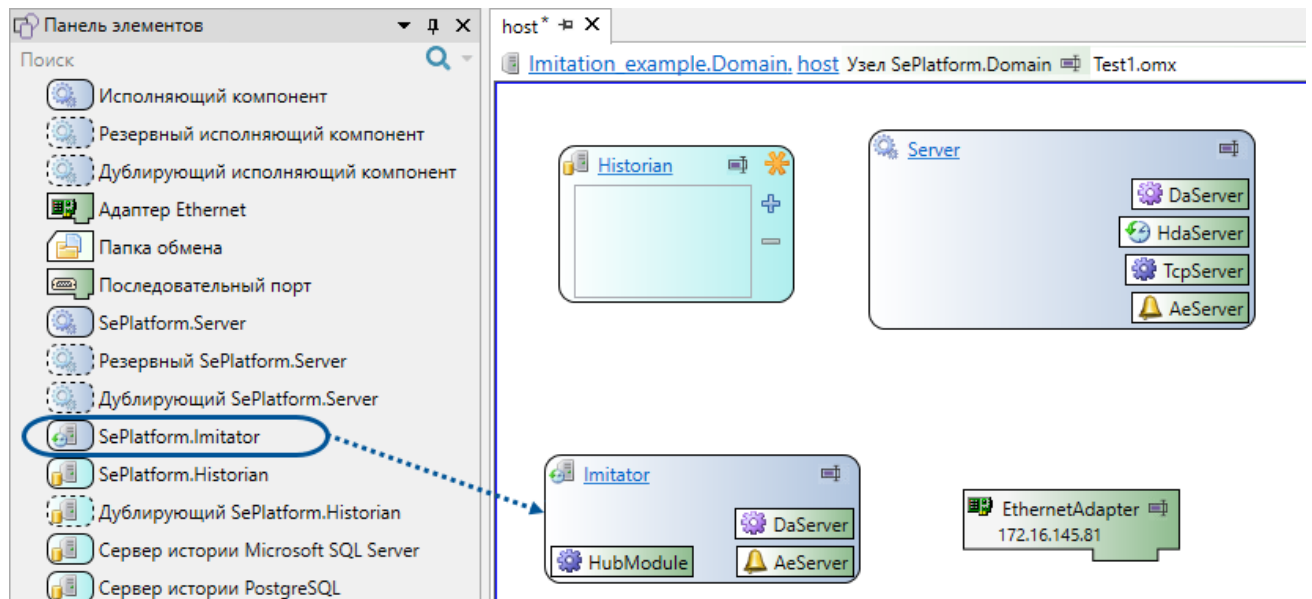
Порядок создания проекта и конфигурирование SePlatform.Data Server и SePlatform.Historian описаны в документации на SePlatform.Development Studio (Руководство пользователя: разделы «Знакомство с SePlatform.Development Studio. Создание простого проекта» и «Разработка проекта. Сохранение значений и событий»).

Далее приведено описание настройки воспроизведения истории в существующем проекте, в котором уже сконфигурирован Узел SePlatform.Domain «host», содержащий SePlatform.Data Server и SePlatform.Historian.



3.1. Добавление SePlatform.Imitator в проект

1. Перейдите в Узел SePlatform.Domain «host» и добавьте элемент SePlatform.Imitator.



2. В файле конфигурации `seplatform.domain.agent.xml` (расположен в папке `C:\Program Files\SePlatform\SePlatform.Domain`) выполните следующие настройки:

2.1. В элемент **InstalledComponents** добавьте дочерний элемент **SePlatform.Data Server** с атрибутами:

- **Name** - произвольное имя;
- **ServiceName** - имя службы `SePlatform.Imitator`.

```
<?xml version="1.0" encoding="utf-8"?>
<SePlatform.Domain.Agent Name="NDA">
  <EntryPointNetAgent Name="ARM" Address="127.0.0.1" Port="1010"/>
  <InstalledComponents>
    <SePlatform.Server Name="Server" ServiceName="SePlatform.Server"/>
    <SePlatform.Server Name="Imitator" ServiceName="SePlatform.Imitator"/>
  </InstalledComponents>
  <Server>
    <Components StoragePath="c:\DomainStorage\cache\server">
      <Component InstalledName="Server" Name="Server" StorageLimitSize="0"
StorageLimitNum="0"/>
    </Components>
  </Server>
  <Options LoggerLevel="2"/>
</SePlatform.Domain.Agent>
```

2.2. В элементе **Server** для дочернего элемента **Components** добавьте элемент **Component** с атрибутами:


- **InstalledName** - значение атрибута **Name** элемента **InstalledComponents**, добавленного на предыдущем шаге;
- **Name** - имя **SePlatform.Imitator**, используемое в **SePlatform.Development Studio**.

```
<?xml version="1.0" encoding="utf-8"?>
<SePlatform.Domain.Agent Name="NDA">
  <EntryPointNetAgent Name="ARM" Address="127.0.0.1" Port="1010"/>
  <InstalledComponents>
    <SePlatform.Server Name="Server" ServiceName="SePlatform.Server"
DefaultActivation="1"/>
    <SePlatform.Server Name="Imitator" ServiceName="SePlatform.Imitator"
DefaultActivation="1"/>
  </InstalledComponents>
  <Server>
    <Components StoragePath="c:\DomainStorage\cache\server">
      <Component InstalledName="Server" Name="Server" StorageLimitSize="0"
StorageLimitNum="0"/>
      <Component InstalledName="Imitator" Name="Imitator"
StorageLimitSize="0" StorageLimitNum="0"/>
    </Components>
  </Server>
  <Options LoggerLevel="2"/>
</SePlatform.Domain.Agent>
```

3.2. Настройки БД

- В **SePlatform.Historian** добавьте и настройте БД.
- В **SePlatform.Data Server** укажите БД, в которые будут сохраняться значения.

3.2.1. Добавление БД в SePlatform.Historian

1. Перейдите в Узел **SePlatform.Domain «host»** и на элементе **SePlatform.Historian** нажмите кнопку  для добавления баз данных.

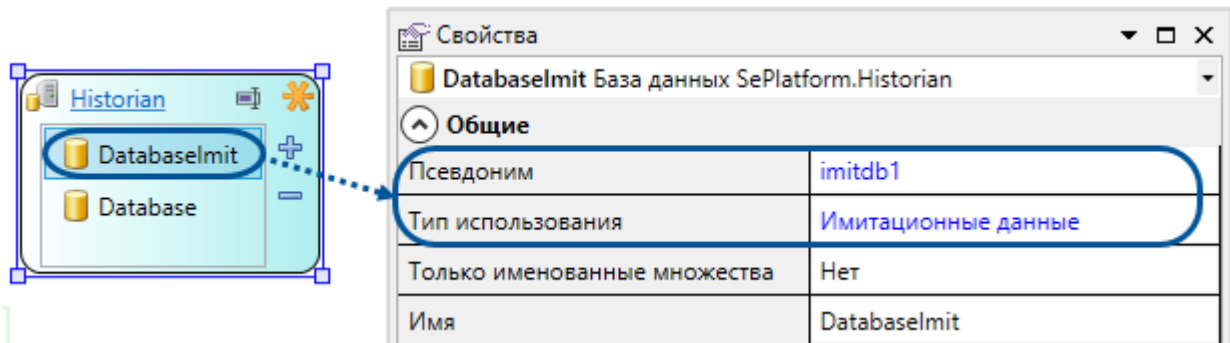
2. Добавьте две базы данных:

- для имитационных данных (будет использоваться в режиме воспроизведения истории);
- для сохранения значений (будет использоваться в режимах перезаписи и дополнения истории).

3. Для первой базы данных укажите значения свойств:

- Псевдоним - имя имитационной базы данных, указанной в настройках SePlatform.Historian: «imitdb1» ([стр. 57](#));
- Тип использования - «Имитационные данные».

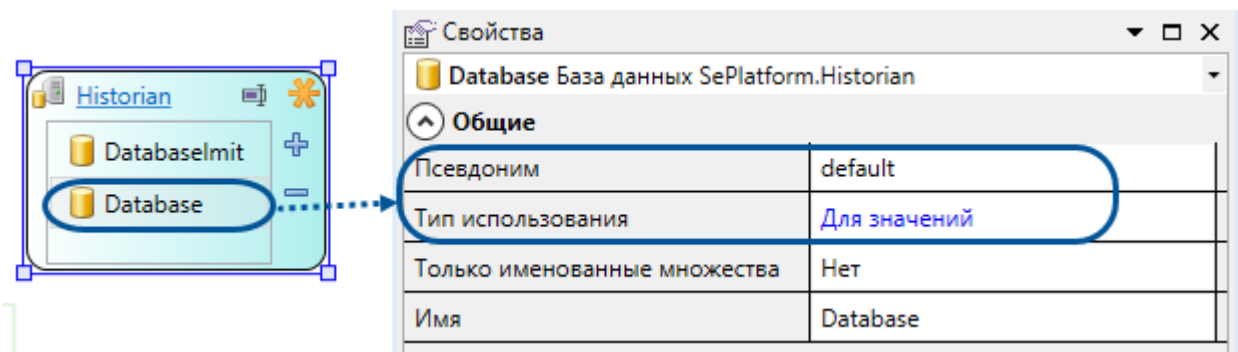
Значение параметра Имя можно указать произвольное.



4. Для второй базы данных укажите значения свойств:

- Псевдоним - имя базы данных, указанной в настройках SePlatform.Historian: «default» ([стр. 57](#));
- Тип использования - «Для значений».

Значение параметра Имя можно указать произвольное.



3.2.2. Сохранение истории в БД

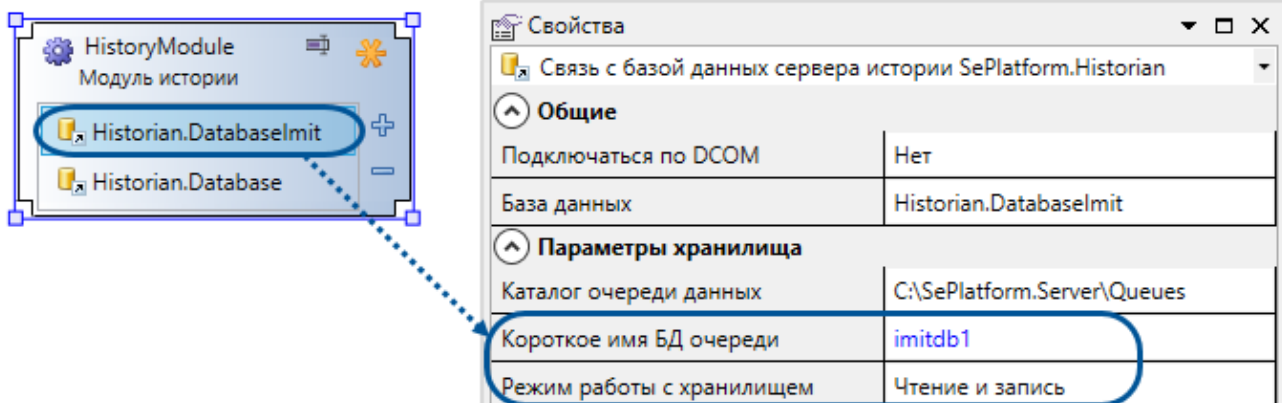
1. Перейдите в SePlatform.Data Server и на элементе Модуль истории нажмите кнопку для добавления баз данных, в которые будет записываться история.

2. Добавьте две базы данных:

- для имитационных данных;
- для сохранения значений.

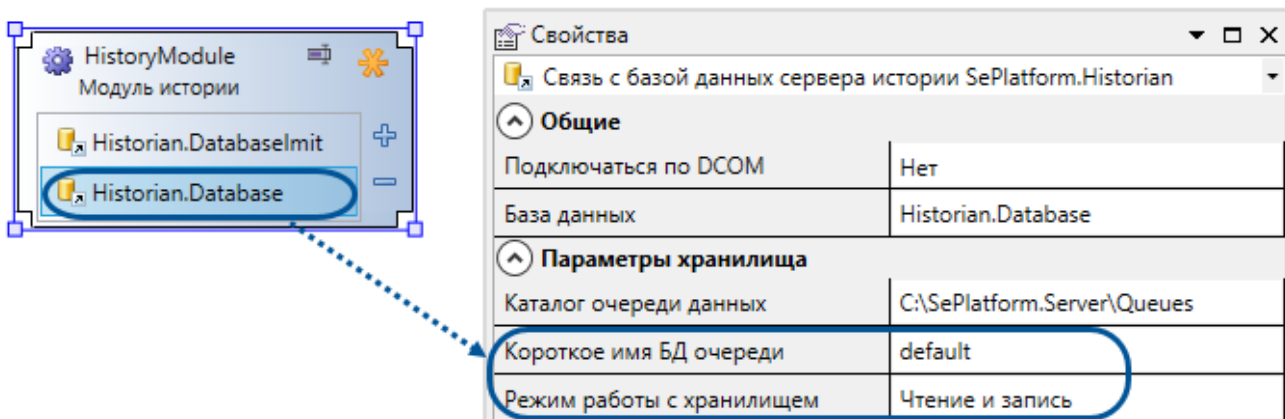
3. Для имитационной БД укажите значения свойств:

- Короткое имя БД очереди - имя имитационной базы данных, в которую будет записываться история - «imitdb1»;
- Режим работы с хранилищем - «Чтение и запись».



4. Для БД сохранения значений укажите значения свойств:

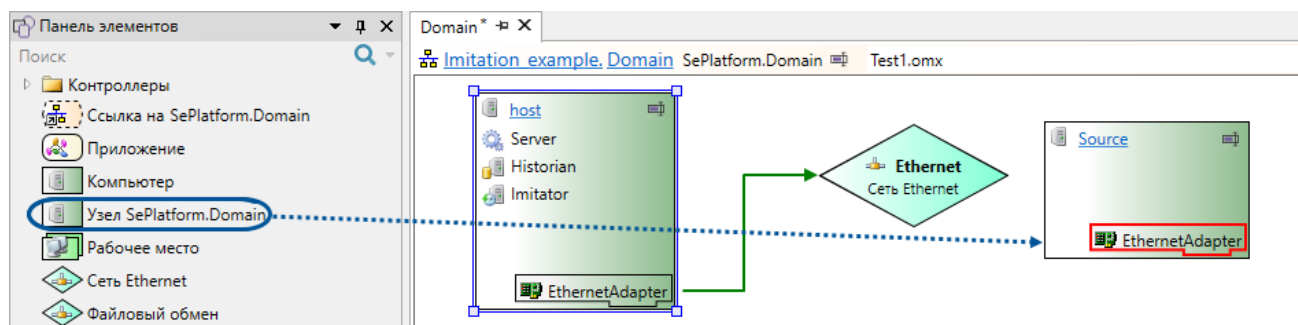
- Короткое имя БД очереди - имя базы данных, в которую будет записываться история - «default»;
- Режим работы с хранилищем - «Чтение и запись».



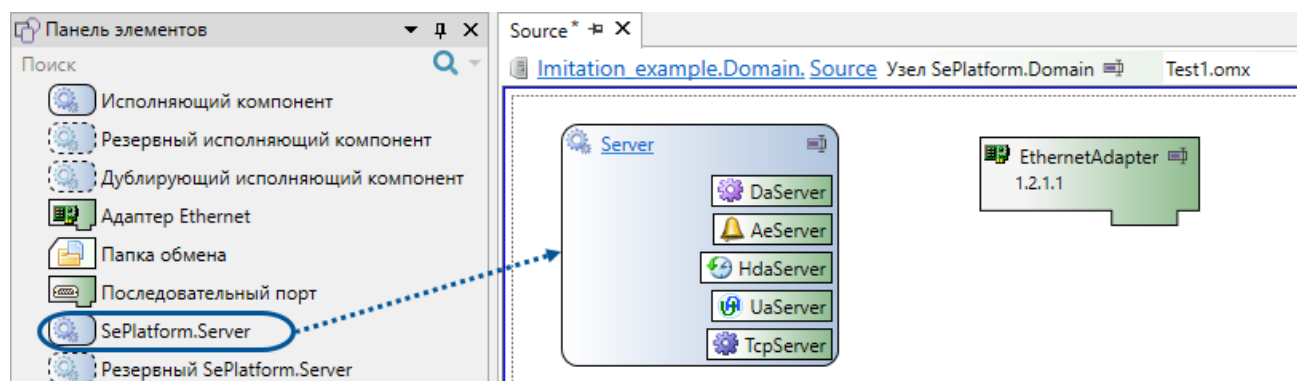
3.3. Настройка источника данных

Настройте источник, от которого SePlatform.Data Server будет получать данные. Источником данных может быть ПЛК или сервер. Далее приведена настройка источника, которым является другой SePlatform.Data Server. Обмен данными между SePlatform.Data Server и источником будет выполняться по TCP:

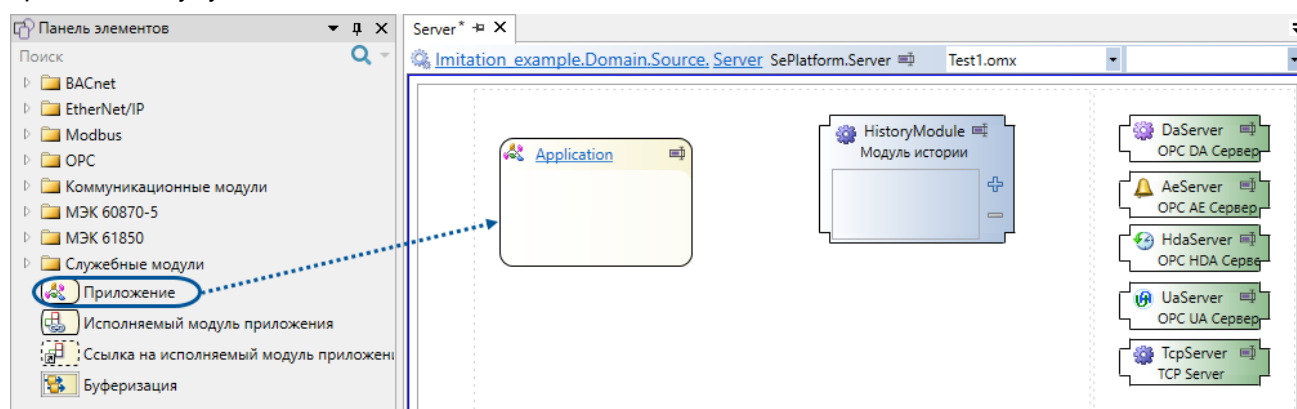
1. Перейдите в SePlatform.Domain и добавьте элемент Узел SePlatform.Domain. Задайте имя, например, «Source».



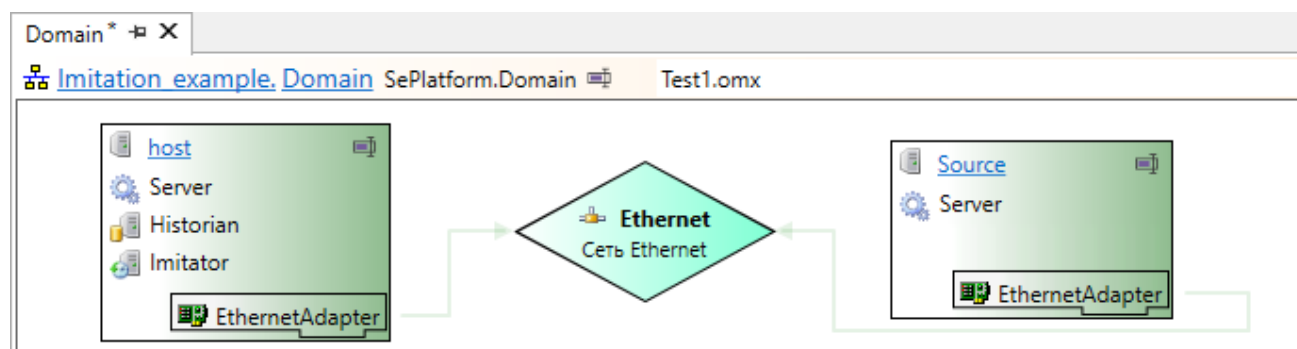
2. Перейдите в Узел **SePlatform.Domain «Source»**, укажите адрес Адаптера Ethernet и добавьте элемент **SePlatform.Data Server**.



3. Перейдите в **SePlatform.Data Server** и добавьте элемент **Приложение**. В дальнейшем в это приложение будут добавлены логические типы источника.

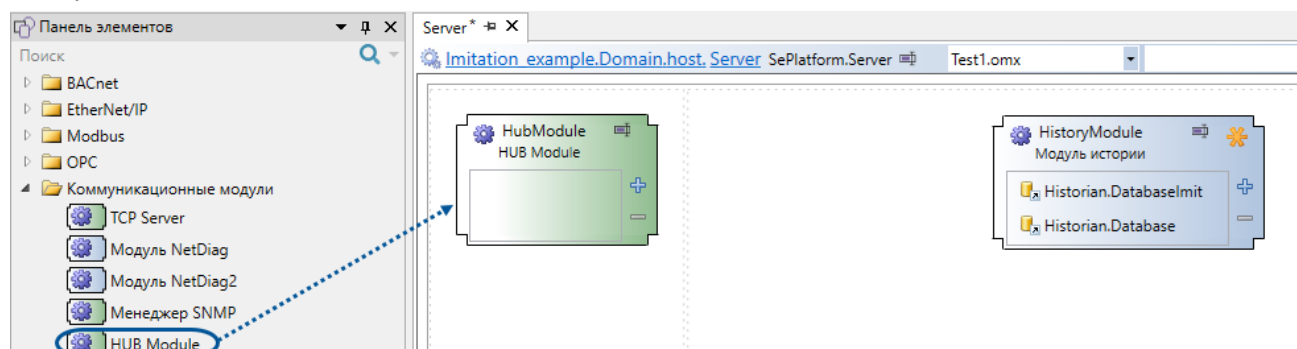


4. Вернитесь в **SePlatform.Domain** и соедините Адаптер Ethernet добавленного источника с Сетью Ethernet.

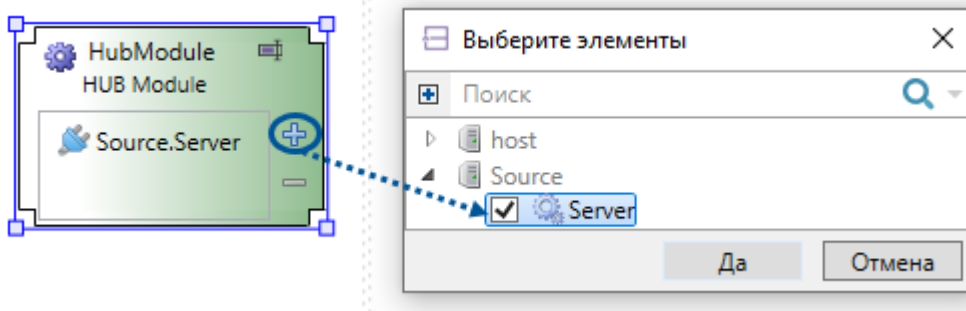


5. Перейдите в Узел **SePlatform.Domain «host»**, содержащий **SePlatform.Data Server**, **SePlatform.Historian** и **SePlatform.Imitator**.

6. Перейдите в **SePlatform.Data Server** и добавьте **HUB Module**.



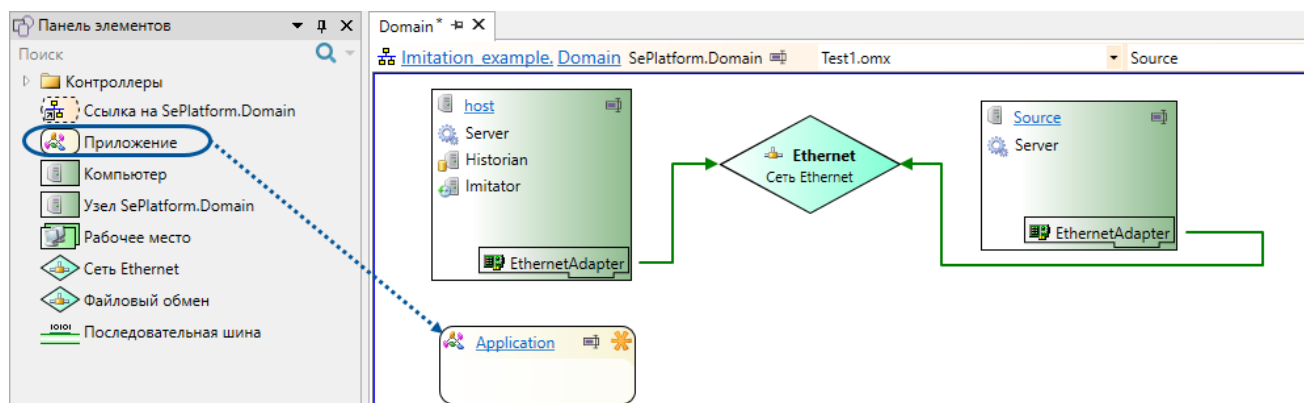
7. Для **HUB Module** укажите ранее настроенный источник данных.



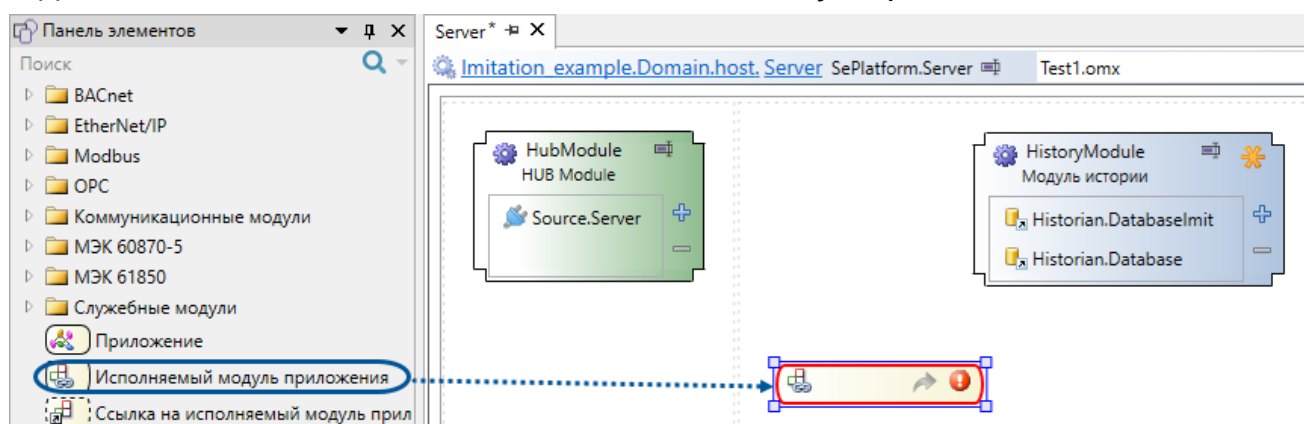
3.4. Настройка приложения SePlatform.Data Server и SePlatform.Imitator

Конфигурации SePlatform.Data Server и SePlatform.Imitator должны быть одинаковыми. Поэтому **Приложение**, описывающее набор параметров, должно быть общее для SePlatform.Data Server и SePlatform.Imitator. Для этого следует разместить **Приложение** в **SePlatform.Domain**, а из SePlatform.Data Server и SePlatform.Imitator сослаться на данное **Приложение**:

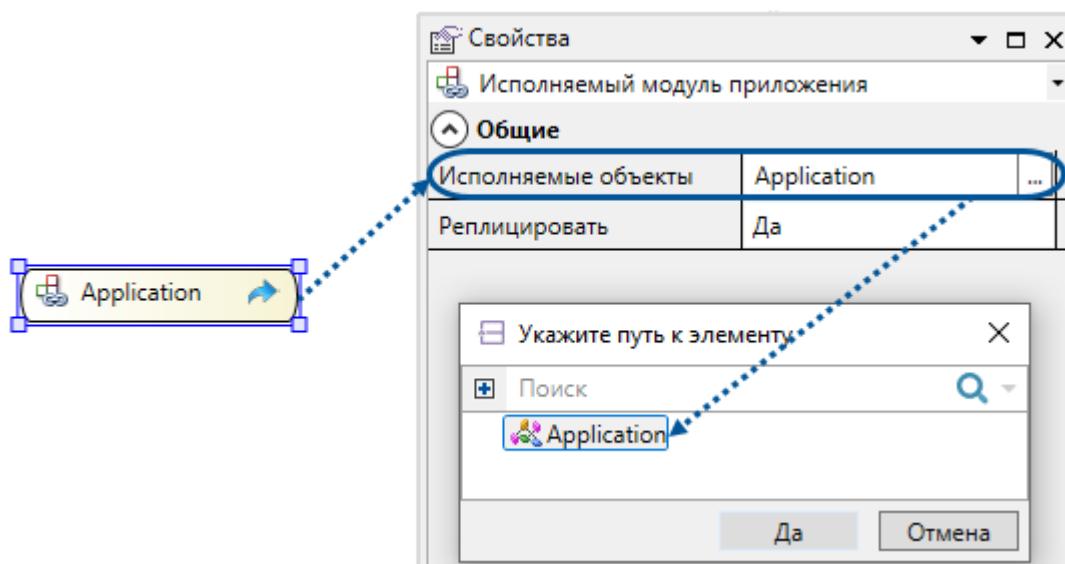
1. Перейдите в **SePlatform.Domain** и добавьте элемент **Приложение**. В дальнейшем в это приложение будут добавлены логические типы SePlatform.Data Server.



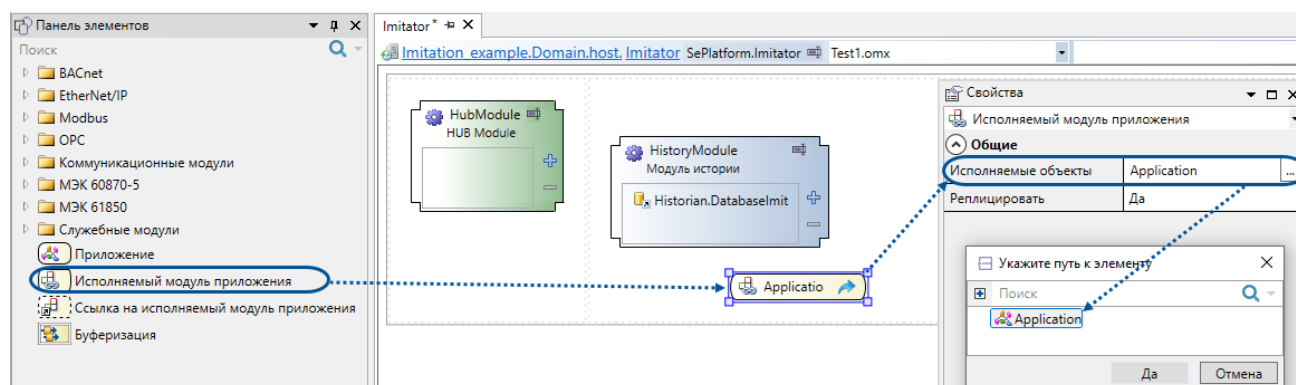
2. Перейдите в Узел **SePlatform.Domain** «host», а затем в **SePlatform.Data Server**.
3. Добавьте в **SePlatform.Data Server** элемент **Исполняемый модуль приложения**.



4. В свойстве **Исполняемые** объектов добавленного элемента укажите добавленное ранее **Приложение**.



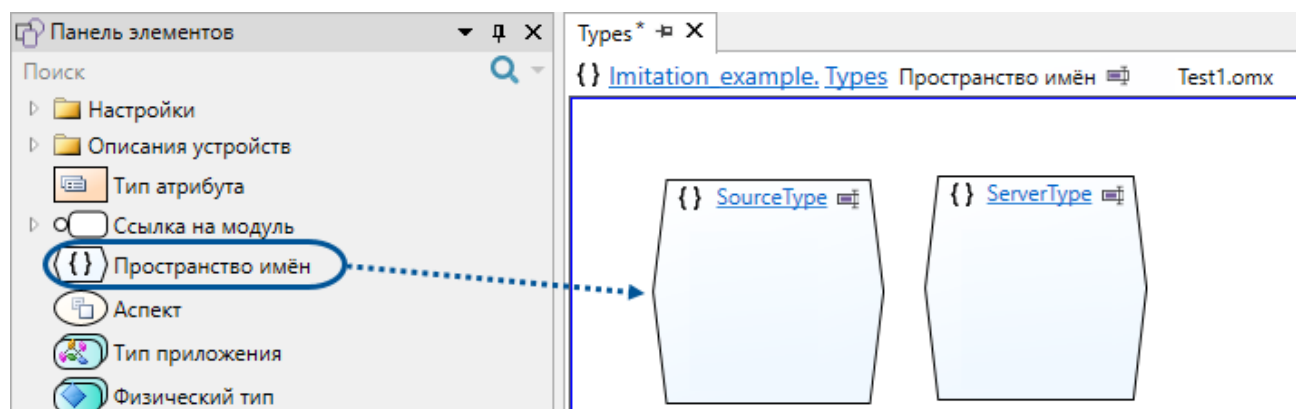
5. Перейдите в **SePlatform.Imitator** и выполните те же действия: добавьте элемент **Исполняемый модуль приложения** и укажите **Приложение**.



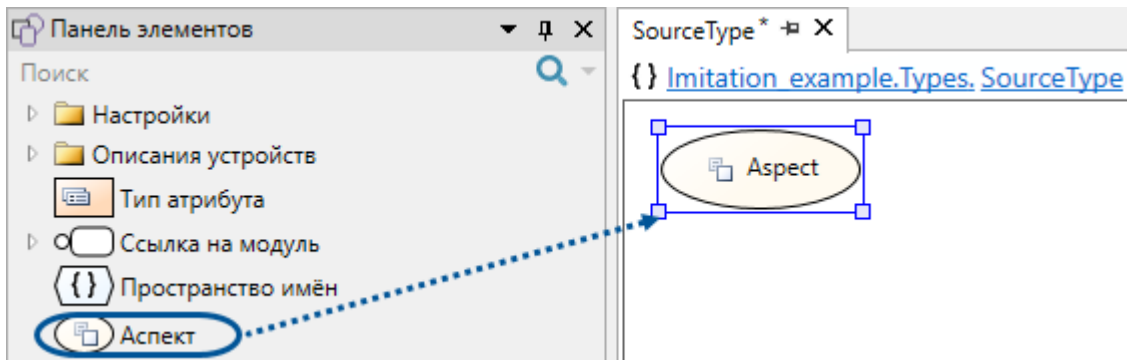
3.5. Настройка логических типов

Для настройки обмена данными между источником и SePlatform.Data Server необходимо настроить соответствующие логические типы.

1. Добавьте в проект элемент **Пространство имён** и задайте **Имя**, например, «Types». Перейдите в пространство имён «Types» и добавьте ещё два элемента **Пространство имён**. Каждому элементу задайте **Имя**, например, «SourceType» и «ServerType». Это будут папки с логическими типами источника и SePlatform.Data Server.



2. В каждое **Пространство имен** добавьте элемент **Аспект**.



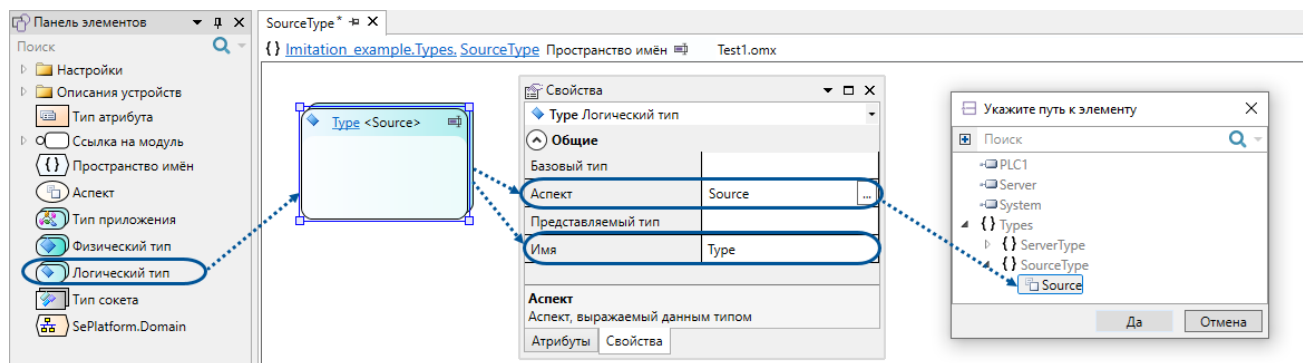
Задайте **Имя** каждому добавленному аспекту:

- в пространстве имён «Source Type» - аспект «Source»;
- в пространстве имён «Server Type» - аспект «Server».

Логический тип источника

1. Перейдите в **Пространство имен** «Source Type» и добавьте элемент **Логический тип**. В свойствах добавленного логического типа укажите:

- **Имя** - имя логического типа источника;
- **Аспект** - добавленный ранее аспект «Source».



2. Перейдите в добавленный **Логический тип** и опишите его структуру: добавьте сигналы необходимых типов, укажите им **Имя** и требуемое **Направление**.

Логический тип SePlatform.Data Server

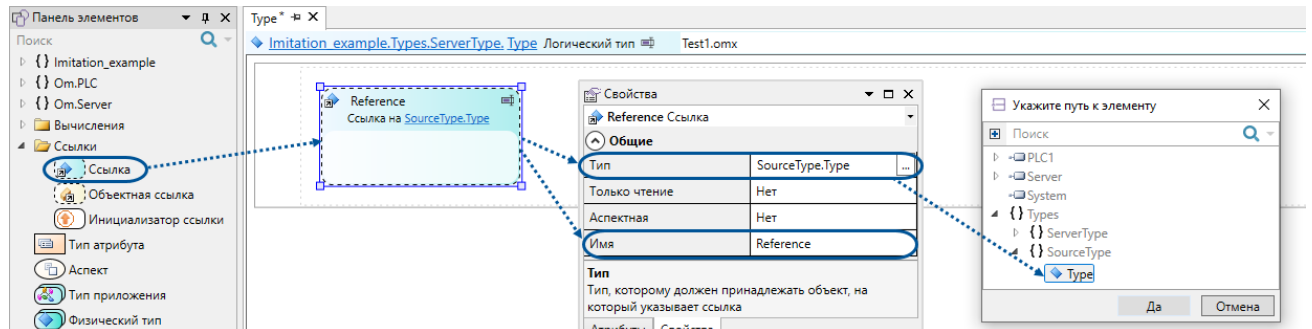
1. Перейдите в **Пространство имен** «Server Type» и добавьте элемент **Логический тип**. В свойствах добавленного логического типа укажите:

- **Имя** - имя логического типа SePlatform.Data Server;
- **Аспект** - добавленный ранее аспект «Server»;
- **Представляемый тип** - описанный ранее логический тип источника.

Добавленный **Логический тип** будет представлять описанный ранее логический тип источника, только со стороны SePlatform.Data Server.

2. Перейдите в добавленный **Логический** тип и добавьте элемент **Ссылка**. В свойствах добавленной ссылки укажите:

- **Имя** - имя ссылки;
- **Тип** - описанный ранее логический тип источника.



После указания **Типа** в элементе **Ссылка** отобразится структура логического типа ПЛК. Добавленная ссылка будет связывать логический тип SePlatform.Data Server с логическим типом ПЛК.


3. В контекстном меню ссылки выполните команду **Экспонировать входы и выходы**, в результате чего в логическом типе SePlatform.Data Server будут созданы сигналы с теми же именами и типами, что и в логическом типе ПЛК, а также прорисованы связи.

3.6. Настройки для режима воспроизведения истории

Для работы SePlatform.Imitator в режиме воспроизведения истории выполните следующие настройки:

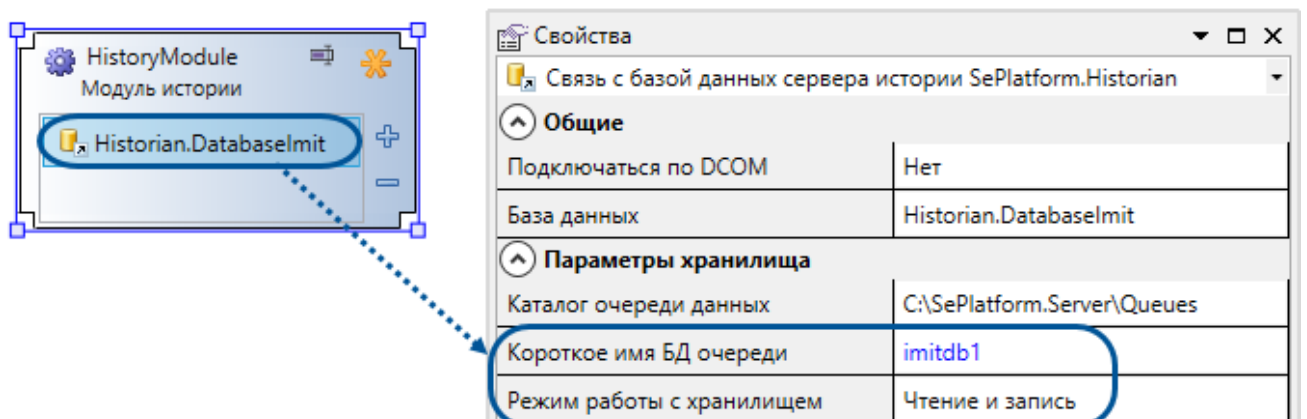
- настройте БД, из которой будет воспроизводиться история;
- настройте сигналы, для которых требуется воспроизводить историю;
- в конфигурацию SePlatform.Imitator добавьте требуемые модули.

3.6.1. Настройка БД для воспроизведения истории

1. В Узле **SePlatform.Domain «host»** перейдите в **SePlatform.Imitator** и на элементе **Модуль истории** нажмите кнопку  для добавления имитационной базы данных SePlatform.Historian, из которой будут загружаться исторические данные для дальнейшего воспроизведения истории.

2. Для добавленной имитационной БД укажите значения свойств:

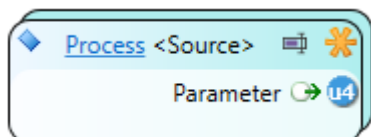
- **Короткое имя БД очереди** - имя имитационной базы данных, из которой будет читаться история для воспроизведения - «**imitdb1**»;
- **Режим работы с хранилищем** - «**Чтение и запись**».



3.6.2. Настройка сигналов для воспроизведения истории

Например, пусть требуется воспроизводить историю изменения значений сигнала «Process.Parameter». Для этого выполните следующие настройки:

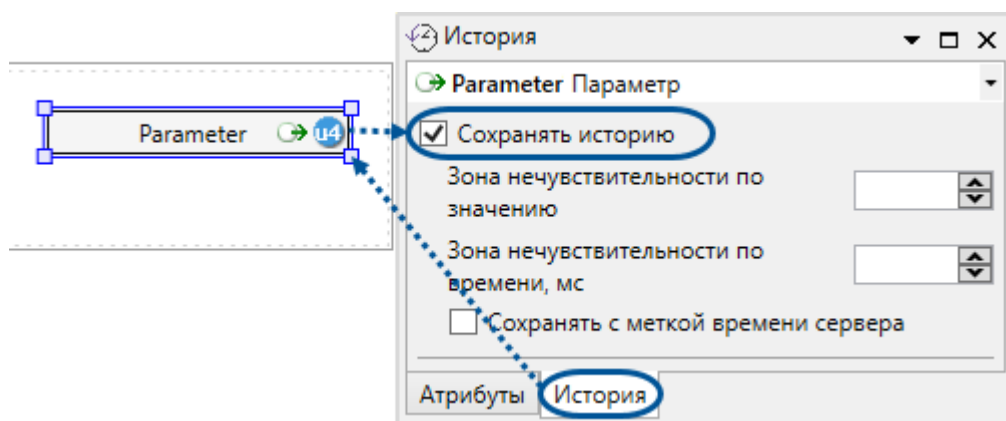
1. Перейдите в **Пространство имен «SourceType»** и добавьте **Логический** тип источника - «Process» ([стр. 18](#)).
2. Перейдите в добавленный тип «Process» и опишите его структуру ([стр. 18](#)). Данный логический тип имеет один выходной параметр типа Uint4.



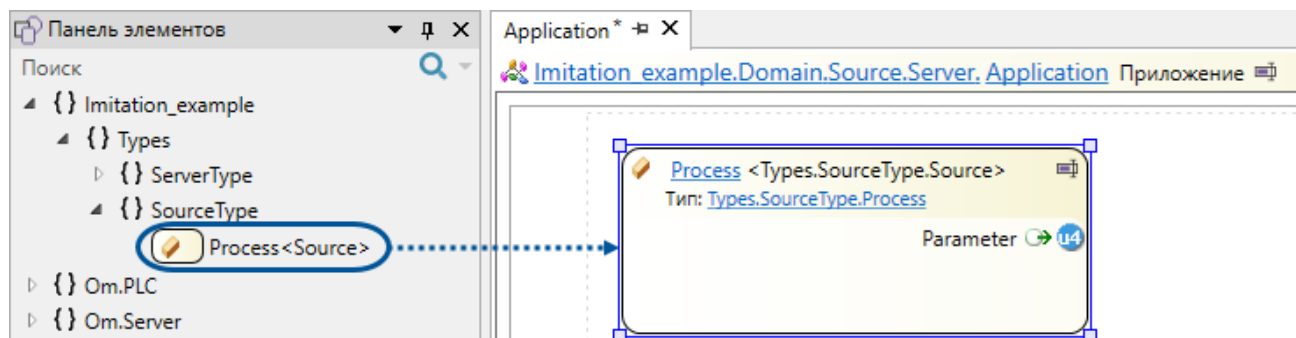
3. Перейдите в **Пространство имен «ServerType»** и добавьте **Логический** тип для представления логического типа источника, только со стороны SePlatform.Data Server. Например, «Process».
4. Перейдите в добавленный логический тип «Process» и свяжите его с логическим типом «Process» источника.



5. Для появившегося сигнала «Parameter» настройте сохранение истории. Для этого выделите сигнал «Parameter» и на вкладке **История** установите флаг **Сохранять историю**.

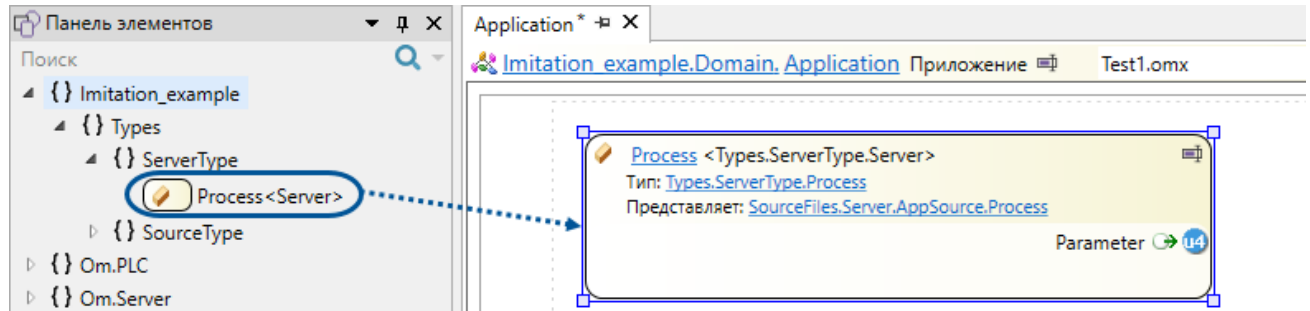


6. Перейдите в источник «Source», а затем в приложение источника «Application», и добавьте экземпляр логического типа источника «Process».




7. Перейдите в Узел **SePlatform.Domain «host»**, а затем в приложение **«Application»**, и добавьте экземпляр логического типа **SePlatform.Data Server «Process»**.

Добавленному типу на вкладке **Свойства** укажите **Представляемый объект** - экземпляр логического типа источника **«Process»**, ранее добавленный в приложение источника.

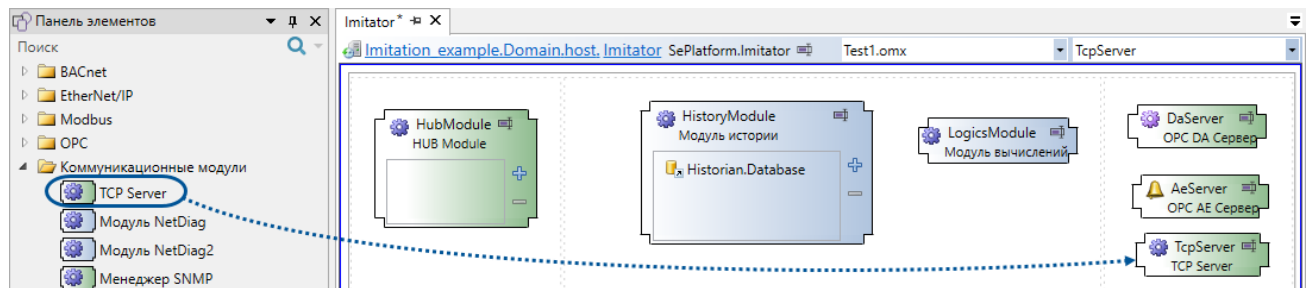


8. Инициализируйте ссылку:

- вызовите контекстное меню и выполните команду **Инициализировать все ссылки**;
- нажмите кнопку  на панели инструментов или клавишу «F5» на клавиатуре.

3.6.3. Требуемые модули SePlatform.Imitator

1. Перейдите в Узел **SePlatform.Domain «host»**, а затем в **SePlatform.Imitator**.
2. Чтобы SePlatform.Imitator предоставлял имитационные данные по TCP клиенту, например **SePlatform.HMI.Trends**, добавьте модуль **TCP Server**.




3.7. Настройки для режима перезаписи истории

Для работы SePlatform.Imitator в режиме перезаписи истории выполните следующие настройки:

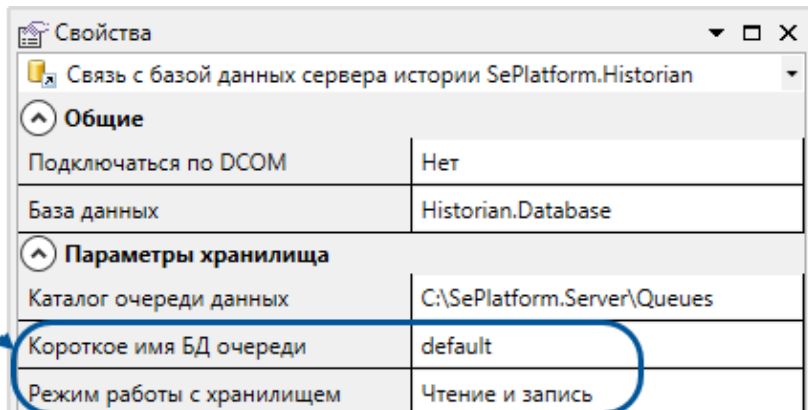
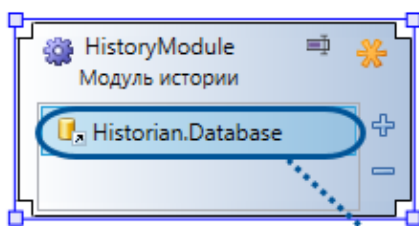
- настройте БД, в которой будет перезаписываться история;
- настройте сигналы, для которых требуется перезаписывать историю;
- в конфигурацию SePlatform.Imitator добавьте требуемые модули.

3.7.1. Настройка БД для перезаписи истории

1. В Узле **SePlatform.Domain «host»** перейдите в **SePlatform.Imitator** и на элементе **Модуль истории** нажмите кнопку  для добавления базы данных, в которую будут перезаписываться исторические данные.

2. Для добавленной базы данных укажите значения свойств:

- Короткое имя БД очереди - имя базы данных, в которую будет перезаписываться история - «default»;
- Режим работы с хранилищем - «Чтение и запись».



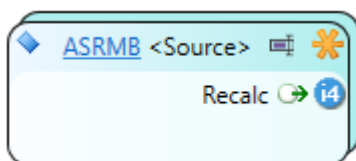
3.7.2. Настройка сигналов для перезаписи истории

Например, пусть значение сигнала «ASRMB.Item» вычисляется на основе значения сигнала «ASRMB.Recalc» по формуле:

$$\text{ASRMB.Item} = \text{ASRMB.Recalc} + 10$$

Требуется пересчитать и перезаписать в SePlatform.Historian значения сигнала «ASRMB.Item» на основе новых значений сигнала «ASRMB.Recalc». Для этого выполните следующие настройки:

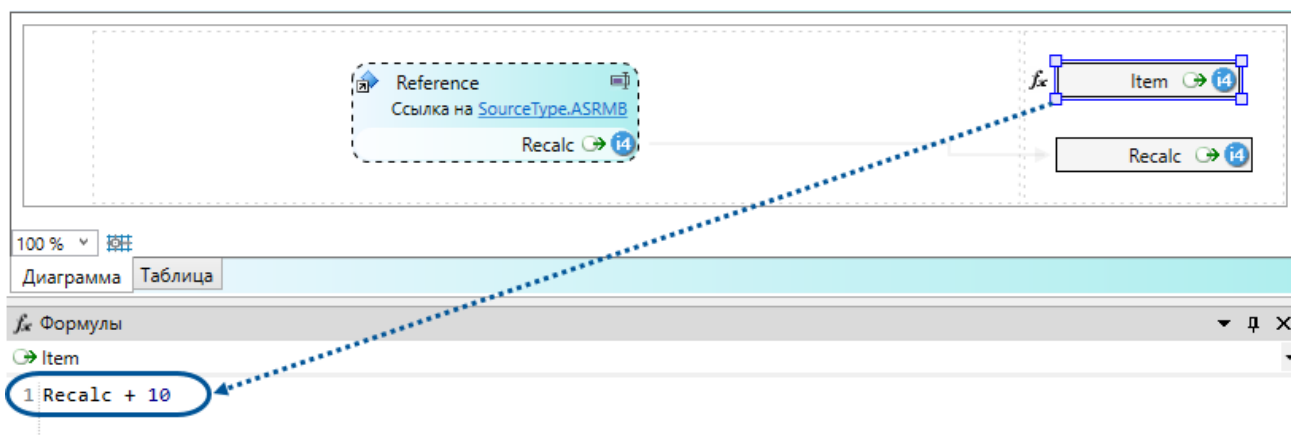
1. Перейдите в **Пространство имен «SourceType»** и добавьте **Логический** тип источника «ASRMB» ([стр. 18](#)).
2. Перейдите в добавленный тип «ASRMB» и опишите его структуру ([стр. 18](#)). Данный логический тип имеет один выходной параметр типа Int4.



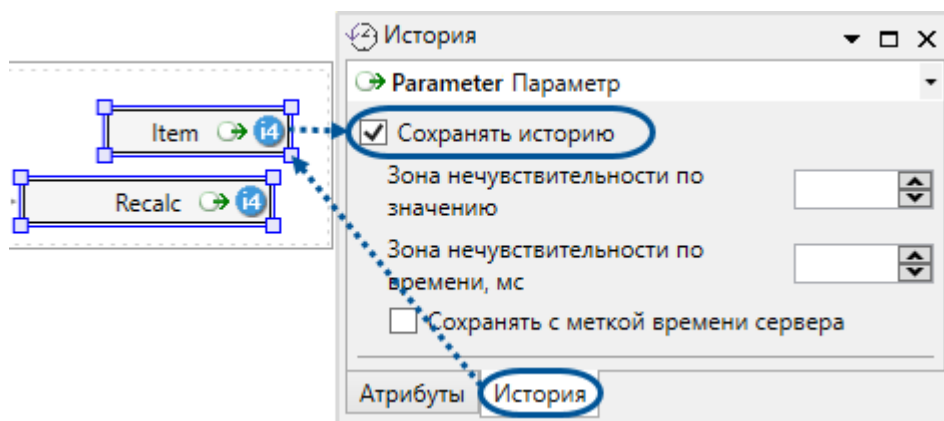
3. Перейдите в **Пространство имен «ServerType»** и добавьте **Логический** тип для представления логического типа источника, только со стороны SePlatform.Data Server. Например, «ASRMB».
4. Перейдите в добавленный логический тип «ASRMB» и свяжите его с логическим типом «ASRMB» источника.



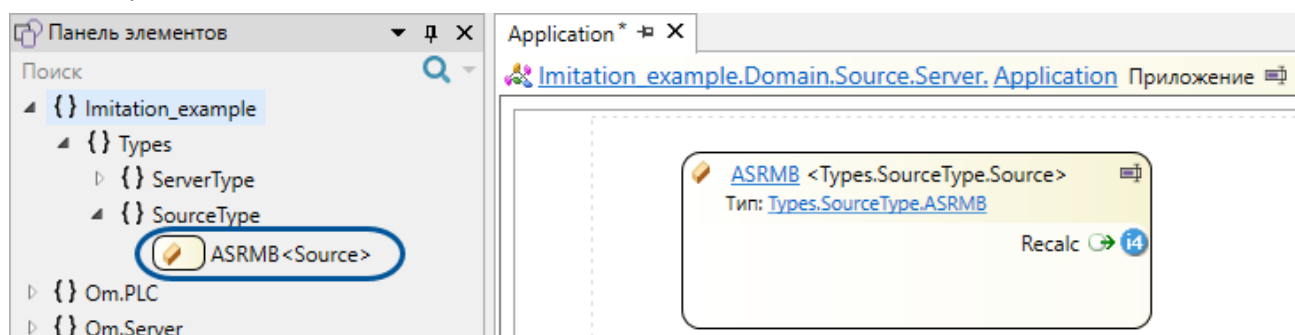
5. В логический тип «ASRMB» добавьте параметр «Item» типа Int4 и на вкладке **Формулы** укажите формулу для расчёта значения.



6. Для параметров «Item» и «Recalc» настройте сохранение истории. Для этого выделите параметры и на вкладке **История** установите флаг **Сохранять историю**.

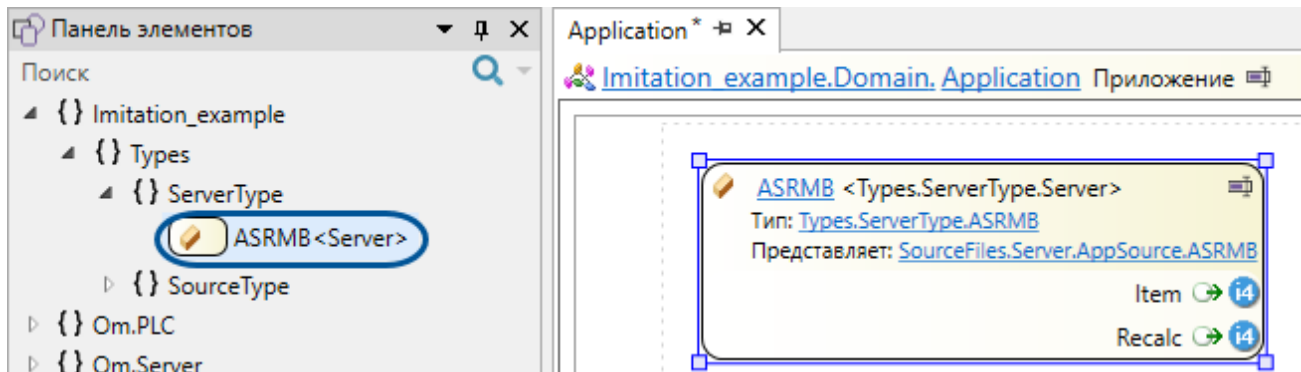


7. Перейдите в источник «Source», а затем в приложение источника «Application», и добавьте экземпляр логического типа источника «ASRMB».




8. Перейдите в Узел **SePlatform.Domain «host»**, а затем в приложение **«Application»**, и добавьте экземпляр логического типа **SePlatform.Data Server «ASRMB»**.

Добавленному типу на вкладке **Свойства** укажите **Представляемый объект** - экземпляр логического типа источника **«ASRMB»**, ранее добавленный в приложение источника.

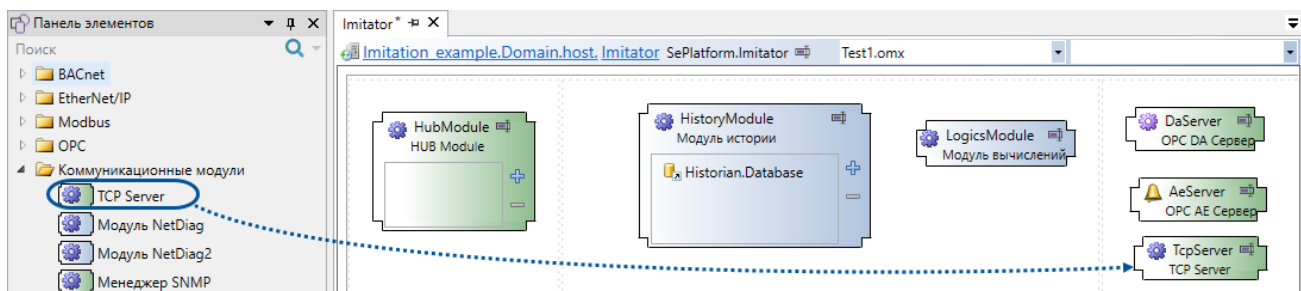


9. Инициализируйте ссылку:

- вызовите контекстное меню и выполните команду **Инициализировать все ссылки**;
- нажмите кнопку  на панели инструментов или клавишу «F5» на клавиатуре.

3.7.3. Требуемые модули SePlatform.Imitator

1. Перейдите в Узел **SePlatform.Domain «host»**, а затем в **SePlatform.Imitator**.
2. Чтобы **SePlatform.Imitator** предоставлял данные по TCP клиенту, например **SePlatform.HMI.Trends**, добавьте модуль **TCP Server**.




3.8. Настройки для режима дополнения истории

Для работы **SePlatform.Imitator** в режиме дополнения истории выполните следующие настройки:

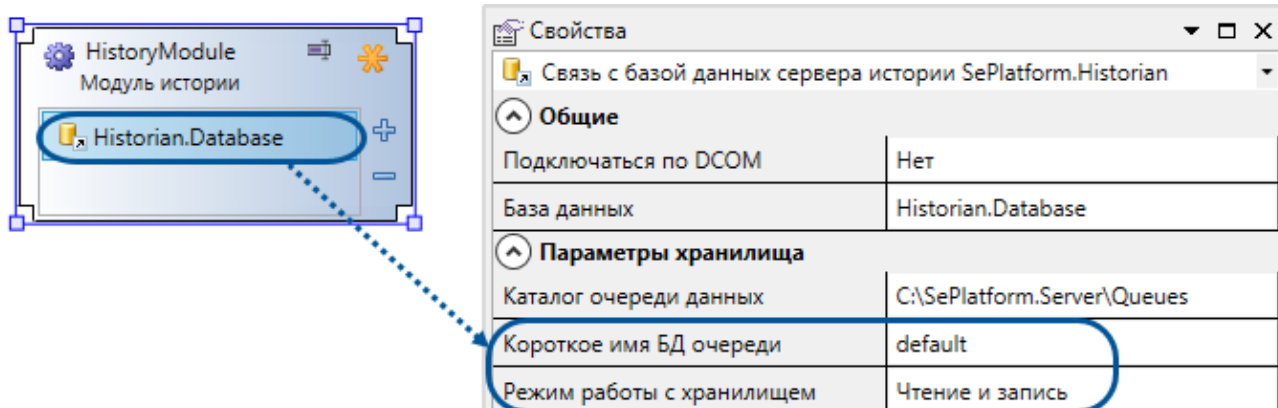
- настройте БД, в которую будет дополняться история;
- настройте сигналы, для которых требуется дополнять историю;
- настройте передачу данных через файловый интерфейс.

3.8.1. Настройка БД для дополнения истории

1. В Узле **SePlatform.Domain «host»** перейдите в **SePlatform.Imitator** и на элементе **Модуль истории** нажмите кнопку  для добавления базы данных, в которую будут дополняться пропущенные исторические данные.

2. Для добавленной базы данных укажите значения свойств:

- **Короткое имя БД очереди** - имя базы данных, в которую будет дополняться история - «default»;
- **Режим работы с хранилищем** - «Чтение и запись».



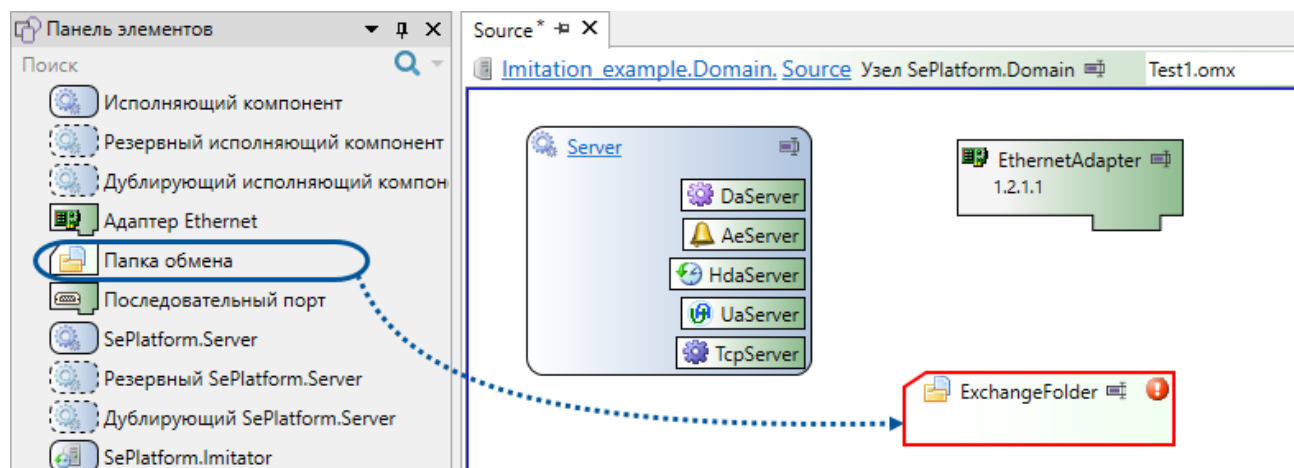
3.8.2. Настройка сигналов для дополнения истории

Сигналам SePlatform.Data Server, для которых требуется дополнять историю, должно быть настроено сохранение значений в SePlatform.Historian. Настройте сигналы аналогично сигналам для воспроизведения истории.

3.8.3. Настройка передачи данных через файловый интерфейс

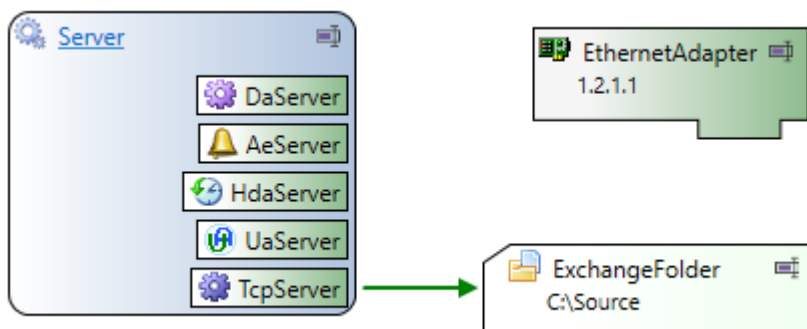
Настройте передачу данных через файловый интерфейс между источником «Source» и SePlatform.Data Server:

1. Перейдите в Узел **SePlatform.Domain «Source»** и добавьте элемент **Папка обмена**.

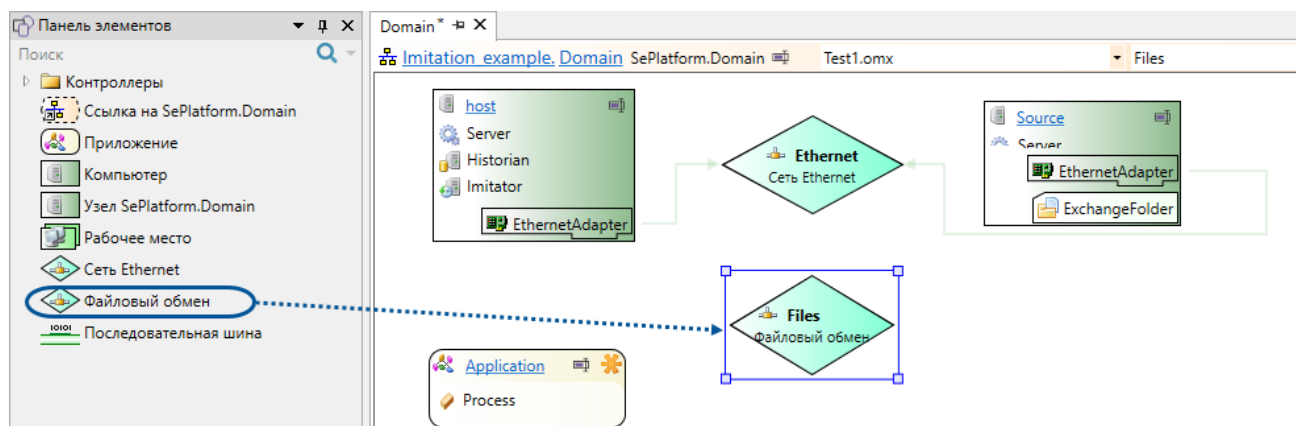


2. На вкладке **Свойства** в значении параметра **Папка** укажите сетевую папку, в которую модуль TCP Server будет сохранять файлы данных.

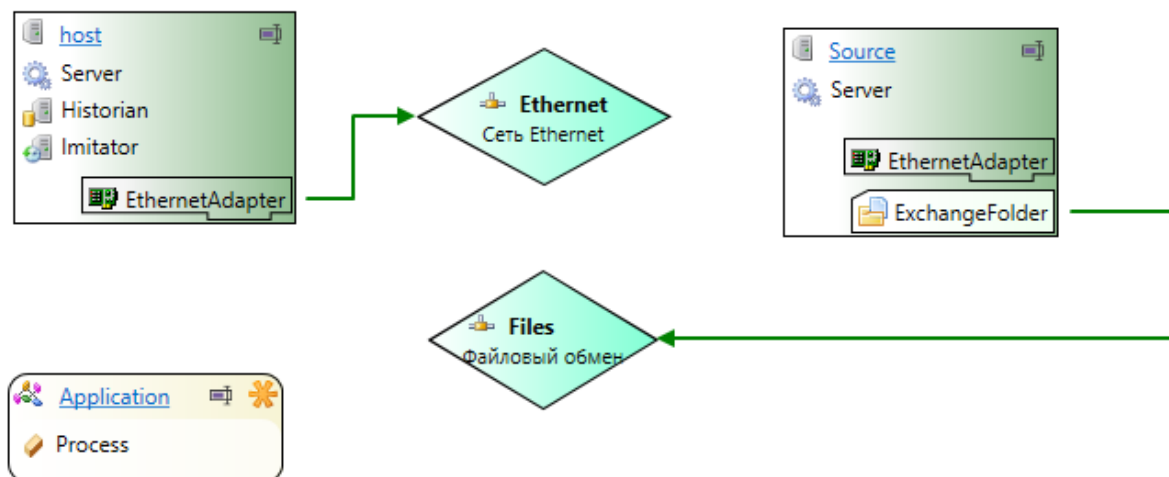
3. Соедините модуль TCP Server с Папкой обмена.



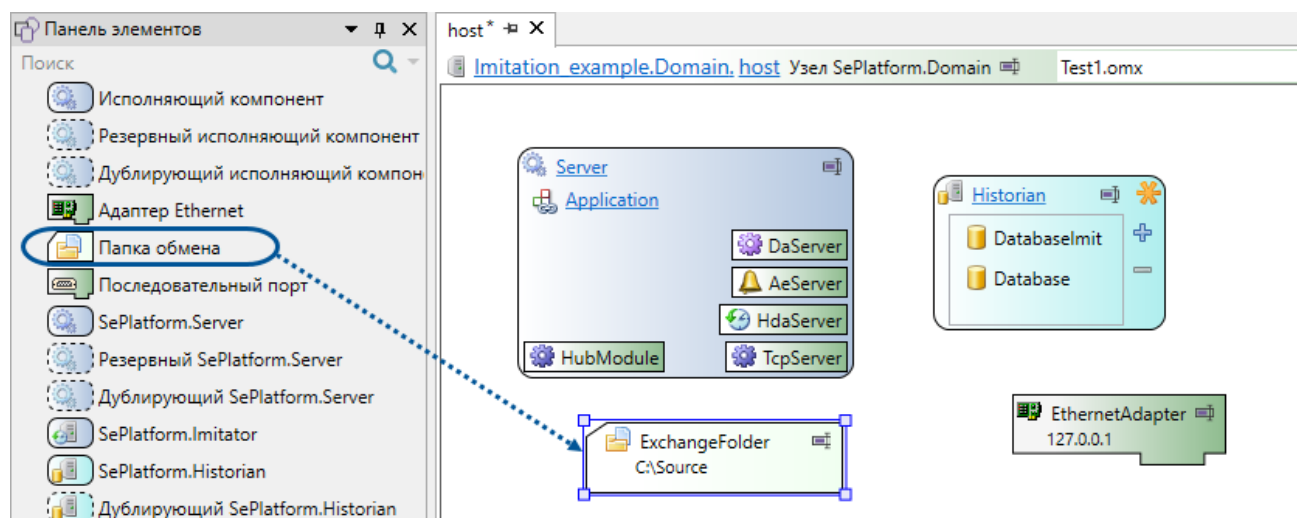
4. Перейдите в SePlatform.Domain и добавьте элемент Файловый обмен.



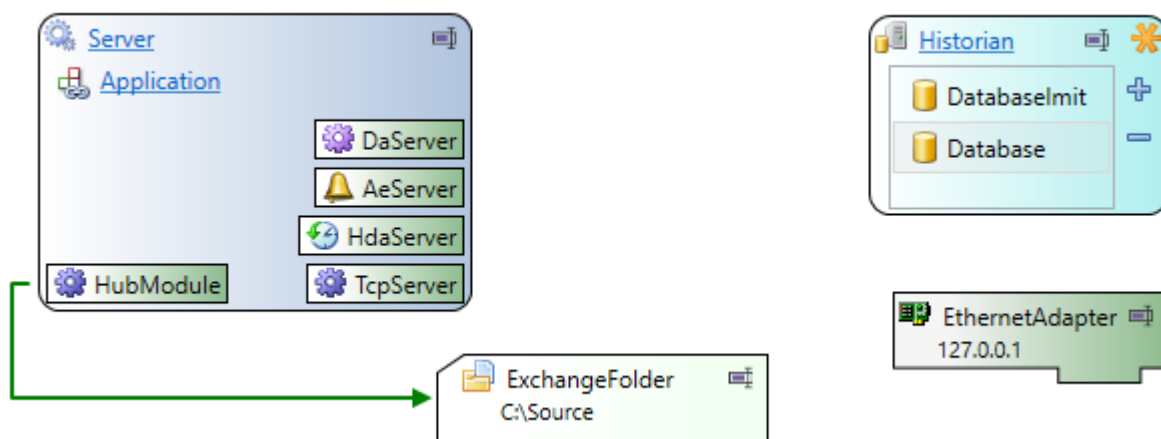
5. Удалите связь между Адаптером Ethernet и Сетью Ethernet, и соедините Папку обмена источника «Source» с элементом Файловый обмен.



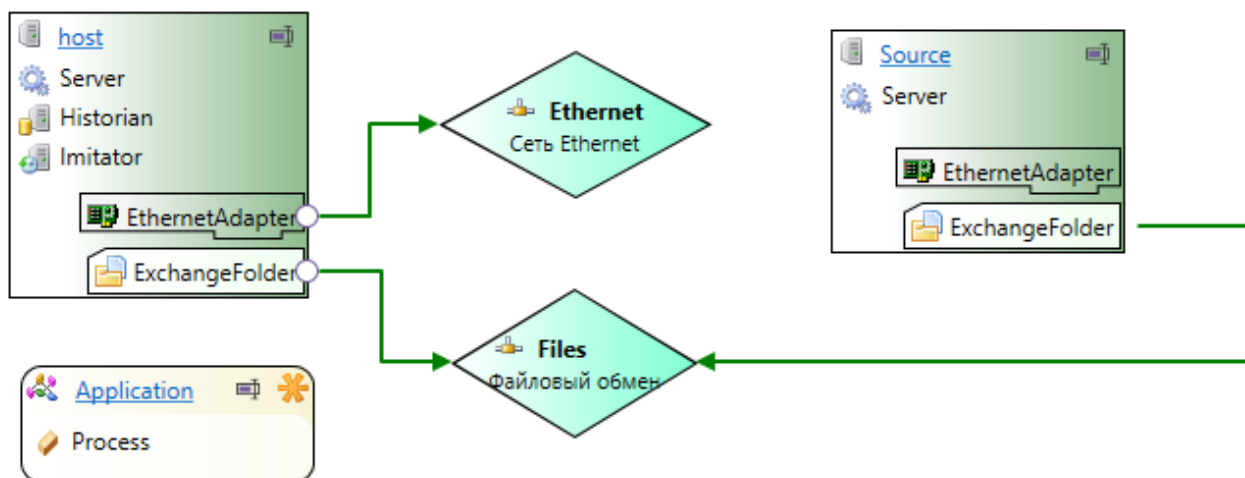
6. Перейдите в Узел **SePlatform.Domain «host»** и добавьте элемент **Папка обмена**. На вкладке **Свойства** в значении параметра **Папка** укажите ту же сетевую папку, что задана для **Папки обмена** источника **«Source»**.



7. Соедините **HUB Module** с **Папкой обмена**. Из этой папки **HUB Module** будет читать файлы данных.



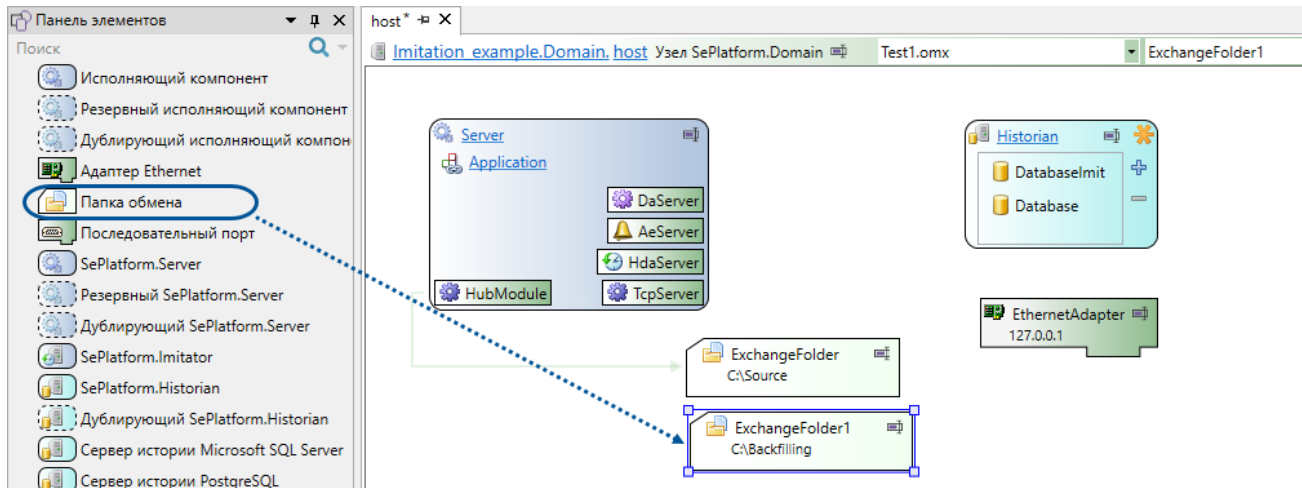
8. Вернитесь в **SePlatform.Domain** и соедините **Папку обмена** Узла **SePlatform.Domain «host»** с элементом **Файловый обмен**.



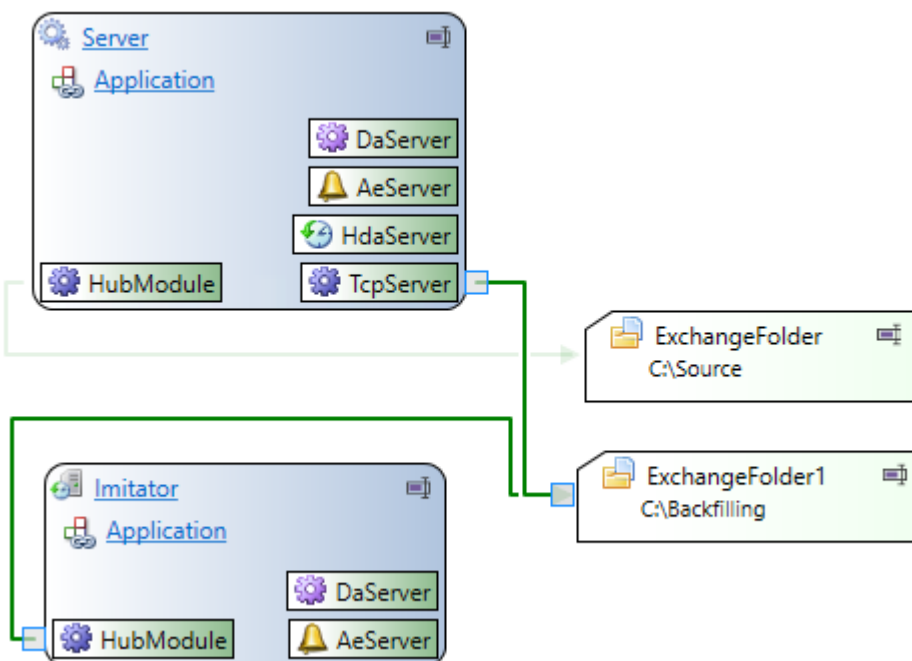
Передача данных через файловый интерфейс между источником «Source» и SePlatform.Data Server настроена.

Теперь нужно настроить передачу данных через файловый интерфейс между SePlatform.Data Server и SePlatform.Imitator:

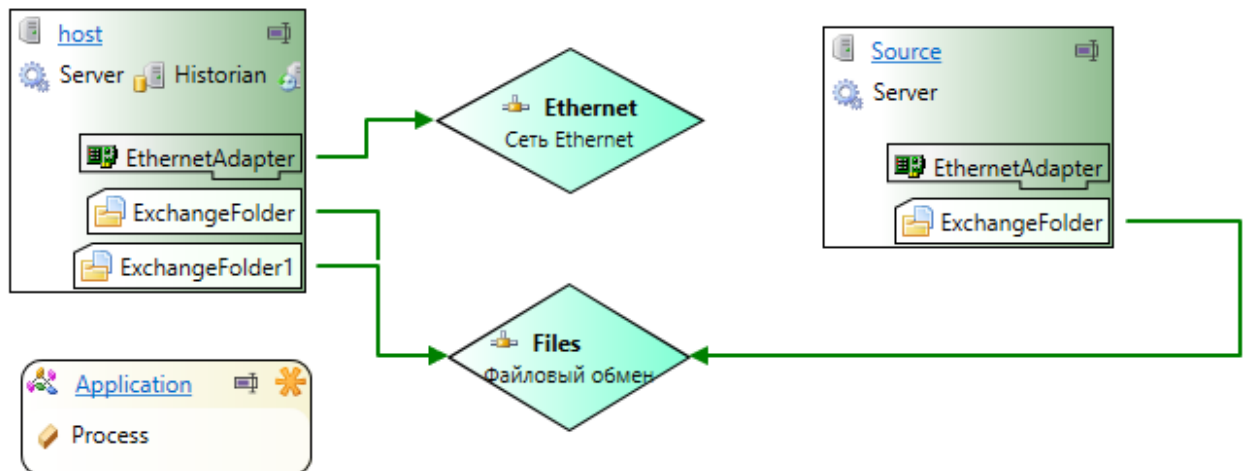
1. Перейдите в Узел **SePlatform.Domain «host»** и добавьте ещё один элемент **Папка обмена**. На вкладке **Свойства** в значении параметра **Папка** укажите сетевую папку, в которую модуль TCP Server SePlatform.Data Server будет сохранять файлы данных, а HUB Module SePlatform.Imitator читать данные.



2. Соедините модуль TCP Server SePlatform.Data Server и HUB Module SePlatform.Imitator с добавленной **Папкой обмена**.



3. Вернитесь в SePlatform.Domain и соедините вторую Папку обмена Узла SePlatform.Domain «host» с элементом **Файловый обмен**.



3.9. Применение конфигураций SePlatform.Data Server и SePlatform.Imitator

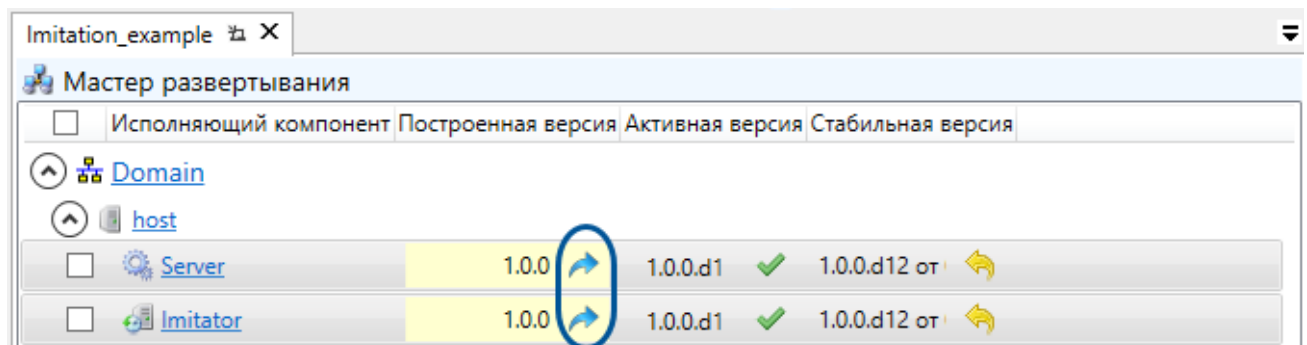
1. Постройте решение. Конфигурации SePlatform.Data Server и SePlatform.Imitator будут построены.



2. Перейдите в Мастер развёртывания.



3. Примените конфигурации к SePlatform.Data Server и SePlatform.Imitator.



4. Настройка SePlatform.Imitator в Конфигураторе



ОБРАТИТЕ ВНИМАНИЕ

Конфигурация SePlatform.Imitator и конфигурация SePlatform.Data Server, выполняющего обработку и запись данных в SePlatform.Historian, должны быть одинаковы.

Чтобы настроить SePlatform.Imitator:

- настройте конфигурацию SePlatform.Data Server для требуемого режима работы SePlatform.Imitator;
- сохраните конфигурацию SePlatform.Data Server;
- загрузите сохраненную конфигурацию SePlatform.Data Server в SePlatform.Imitator.

4.1. Настройка конфигурации для режима воспроизведения истории

Для работы SePlatform.Imitator в режиме воспроизведения истории выполните следующие настройки:

- настройте сигналы, для которых требуется воспроизводить историю;
- в конфигурацию SePlatform.Data Server добавьте требуемые модули;
- настройте сохранение данных в имитационную БД.

4.1.1. Настройка сигналов

Сигналам SePlatform.Data Server, для которых требуется воспроизводить историю, настройте сохранение значений в SePlatform.Historian. Для этого добавьте сигналам свойство **9001 (Historizing)** со значением «True».

Например, требуется воспроизводить историю изменения значений сигнала «Process.Parameter»:

Номер	Имя	Значение	Описание
u4 1	CDT	6	Канонический тип данных
\$ 5000	Address	{ ModuleId=(IEC-104 Master) Protocol...	Адрес параметра
B 9001	Historizing	True	Флаг ведения истории изменений

Добавить Изменить Удалить

4.1.2. Требуемые модули

Добавьте в состав конфигурации следующие модули:

1. TCP Server - модуль будет предоставлять по TCP имитационные данные клиенту, например SePlatform.HMI.Trends.

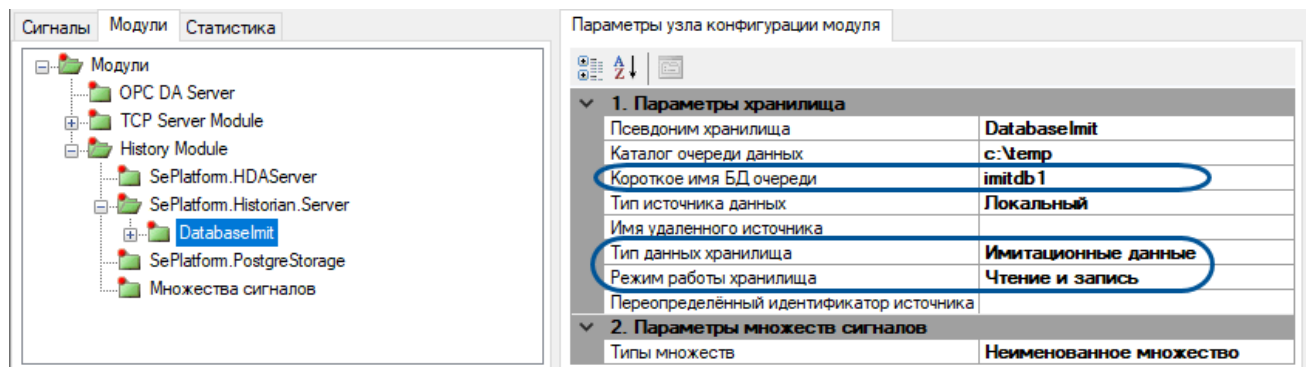
2. OPC DA Server - модуль нужен для управления процессом имитации через служебные сигналы по OPC DA с помощью OPC клиента, например Service - OPCExplorer.
3. History Module - модуль будет получать данные из имитационной базы SePlatform.Historian.

4.1.3. Настройка сохранения данных в имитационную БД

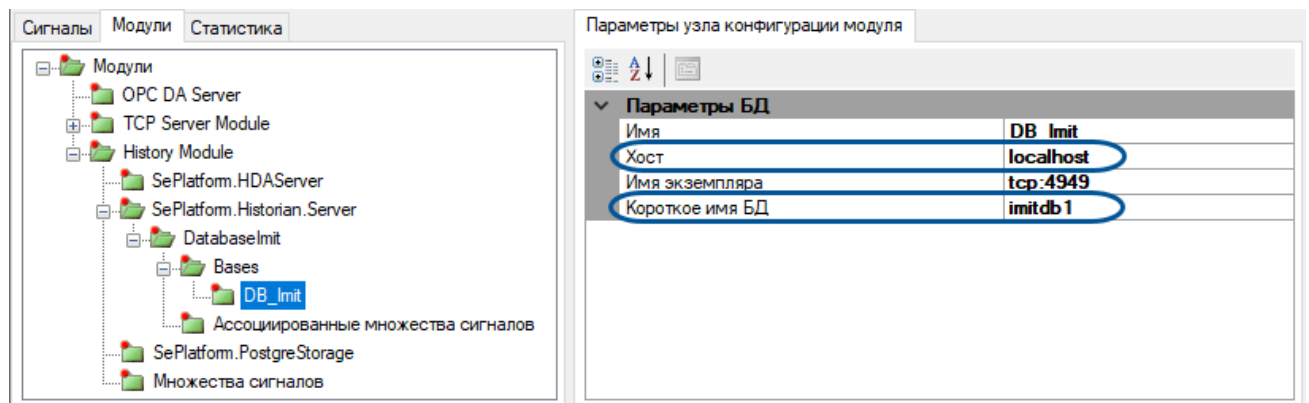
Настройте сохранение данных в имитационную БД SePlatform.Historian ([стр. 57](#)) для дальнейшего воспроизведения истории с помощью SePlatform.Imitator.

Чтобы настроить сохранение данных в имитационную БД SePlatform.Historian:

1. В модуле History Module добавьте хранилище для имитационных данных и укажите его параметры:
 - **Короткое имя БД очереди** - «**imitdb1**» (имя созданной в SePlatform.Historian имитационной базы ([стр. 57](#)));
 - **Тип данных хранилища** - «Имитационные данные»;
 - **Режим работы хранилища** - «Чтение и запись».



2. В хранилище добавьте базу для сохранения имитационных данных и укажите её параметры:
 - **Хост** - IP-адрес SePlatform.Historian;
 - **Короткое имя БД** - «**imitdb1**» (имя созданной в SePlatform.Historian имитационной базы ([стр. 57](#))).



4.2. Настройка конфигурации для режима перезаписи истории

Для работы SePlatform.Imitator в режиме перезаписи истории выполните следующие настройки:

- настройте сигналы, для которых требуется выполнять пересчёт и перезаписывать историю;
- в конфигурацию SePlatform.Data Server добавьте требуемые модули;

- настройте сохранение данных в БД SePlatform.Historian.

4.2.1. Настройка сигналов

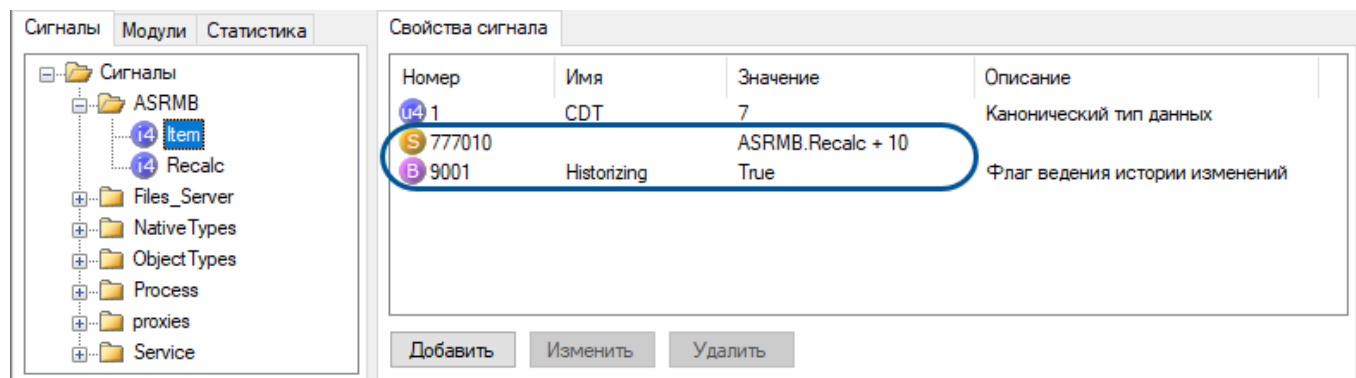
Сигналам SePlatform.Data Server, для которых требуется выполнять пересчёт и перезаписывать историю, настройте логику вычислений и сохранение значений в SePlatform.Historian. Для сохранения значений в SePlatform.Historian добавьте сигналам свойство **9001 (Historizing)** со значением «True».

Например, значение сигнала «ASRMB.Item» вычисляется на основе значения сигнала «ASRMB.Recalc» по формуле:

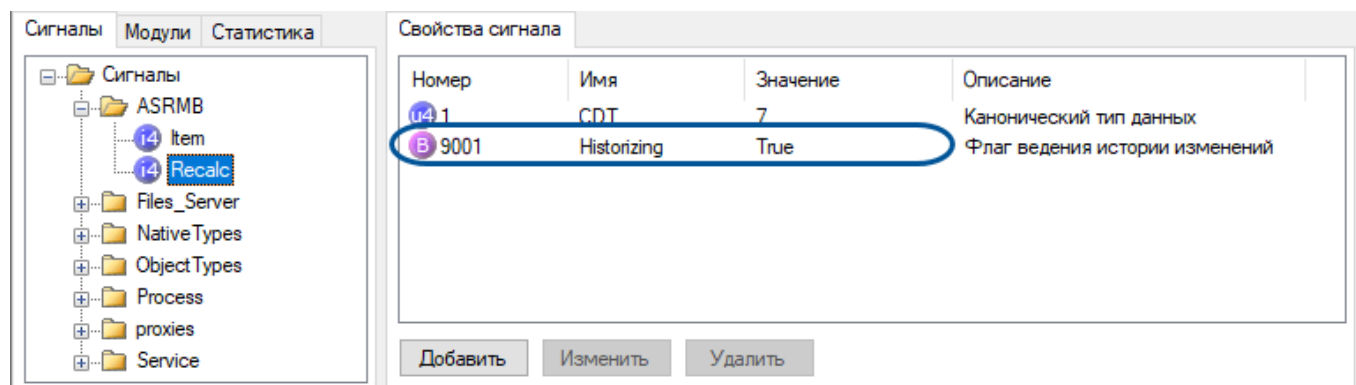
```
ASRMB.Item = ASRMB.Recalc + 10
```

Требуется пересчитать и перезаписать в SePlatform.Historian значения сигнала «ASRMB.Item» на основе новых значений сигнала «ASRMB.Recalc».

Настройки для сигнала «ASRMB.Item»:



Настройки для сигнала «ASRMB.Recalc»:



4.2.2. Требуемые модули

Добавьте в состав конфигурации следующие модули:

1. TCP Server - модуль будет предоставлять по TCP данные клиенту, например SePlatform.HMI.Trends.
2. OPC DA Server - модуль нужен для управления процессом имитации через служебные сигналы по OPC DA с помощью OPC клиента, например Service - OPCExplorer..

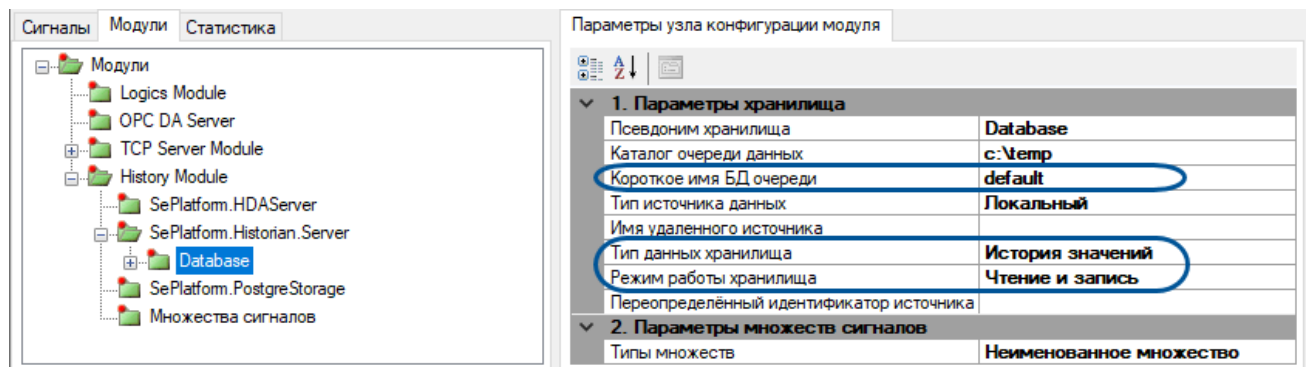
3. Logics Module - в режиме перезаписи истории модуль будет выполнять логическую обработку данных: пересчёт на основе параметров, значения которых задаёт пользователь.
4. History Module - модуль будет записывать пересчитанные значения в БД SePlatform.Historian.

4.2.3. Настройка сохранения данных в БД

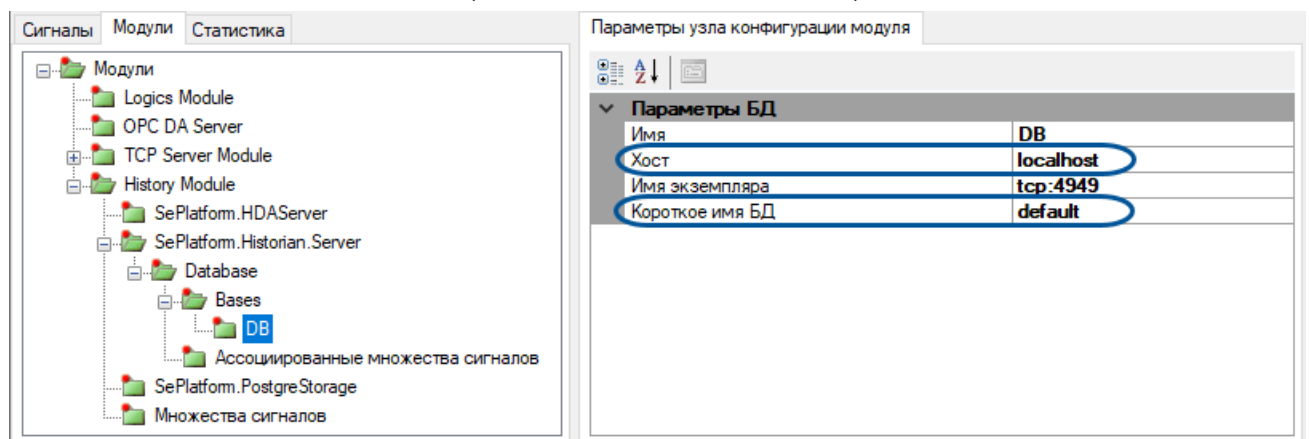
Для работы SePlatform.Imitator в режиме перезаписи истории настройте сохранение данных в БД SePlatform.Historian. В дальнейшем с помощью SePlatform.Imitator данные этой БД могут быть пересчитаны и перезаписаны.

Чтобы настроить сохранение данных в БД SePlatform.Historian:

1. В модуле History Module добавьте хранилище и укажите его параметры:
 - Короткое имя БД очереди - «default» (имя БД в SePlatform.Historian);
 - Тип данных хранилища - «История значений»;
 - Режим работы хранилища - «Чтение и запись».



2. В хранилище добавьте базу для сохранения значений и укажите её параметры:
 - Хост - IP-адрес SePlatform.Historian;
 - Короткое имя БД - «default» (имя БД в SePlatform.Historian).



4.3. Настройка конфигурации для режима дополнения истории

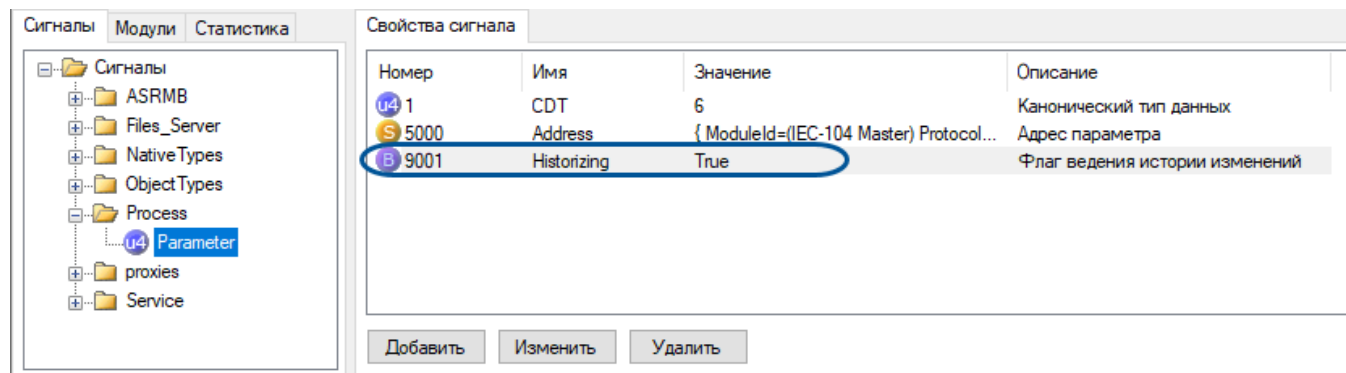
Для работы SePlatform.Imitator в режиме дополнения истории выполните следующие настройки:

- настройте сигналы, для которых требуется дополнять историю;
- в конфигурацию SePlatform.Data Server добавьте требуемые модули;

- настройте сохранение данных в БД SePlatform.Historian;
- настройте файловый интерфейс.

4.3.1. Настройка сигналов

Сигналам SePlatform.Data Server, для которых требуется дополнять историю, настройте сохранение значений в SePlatform.Historian. Для этого добавьте сигналам свойство **9001 (historizing)** со значением «True».



4.3.2. Требуемые модули

Добавьте в состав конфигурации следующие модули:

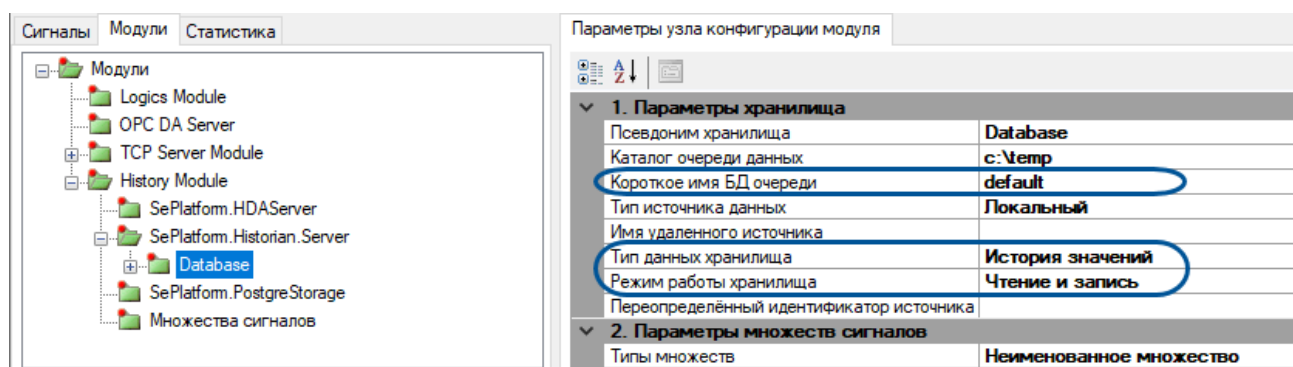
1. OPC DA Server - модуль нужен для управления процессом имитации через служебные сигналы по OPC DA с помощью OPC клиента, например Service - OPCExplorer.
2. HUB Module - режим дополнения пропущенных данных модуль будет получать файлы для дополнения пропущенных исторических данных.
3. History Module - модуль будет записывать пропущенные данные в БД SePlatform.Historian.

4.3.3. Настройка сохранения данных в БД

Для работы SePlatform.Imitator в режиме дополнения пропущенных данных настройте сохранение данных в БД SePlatform.Historian. В дальнейшем с помощью SePlatform.Imitator пропущенные данные этой БД могут быть дополнены.

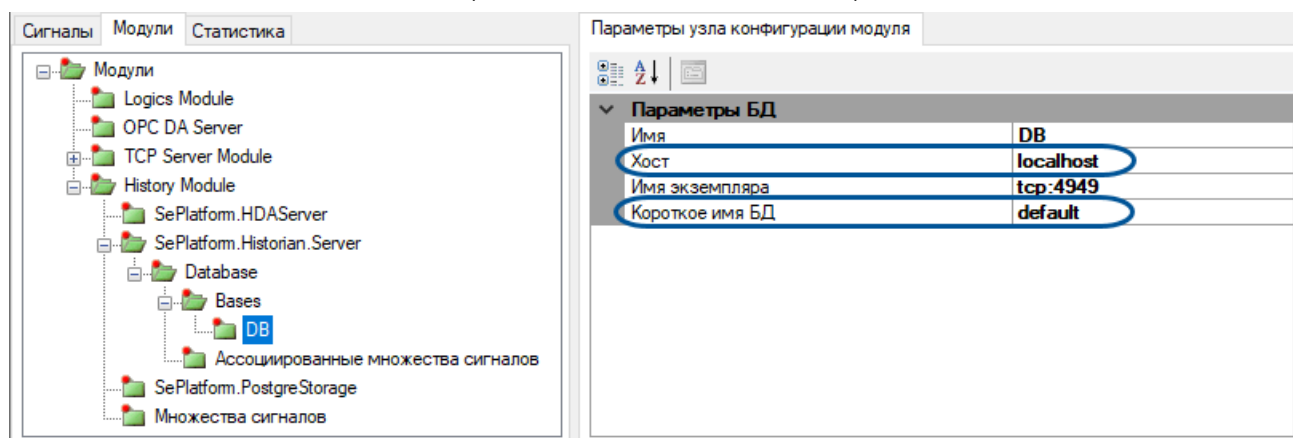
Чтобы настроить сохранение данных в БД SePlatform.Historian:

1. В модуле History Module добавьте хранилище и укажите его параметры:
 - Короткое имя БД очереди - «default» (имя БД в SePlatform.Historian);
 - Тип данных хранилища - «История значений»;
 - Режим работы хранилища - «Чтение и запись».



2. В хранилище добавьте базу для сохранения значений и укажите её параметры:

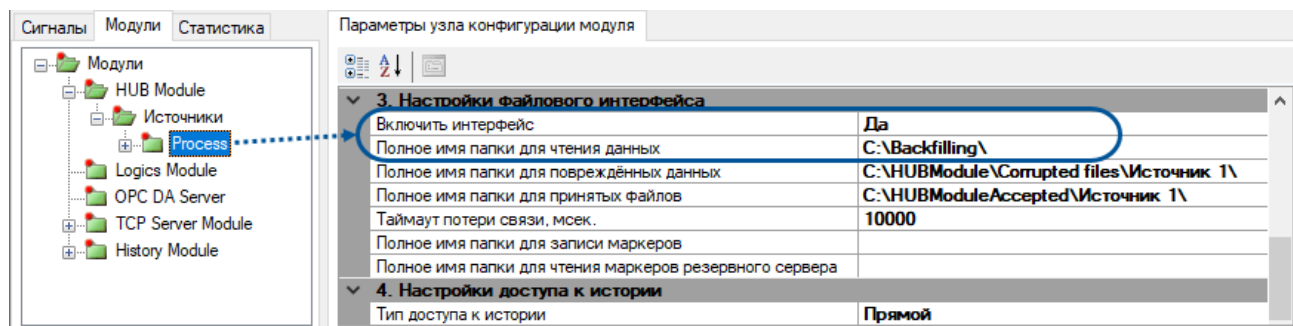
- Хост - IP-адрес SePlatform.Historian;
- Короткое имя БД - «default» (имя БД в SePlatform.Historian).



4.3.4. Настройка файлового интерфейса

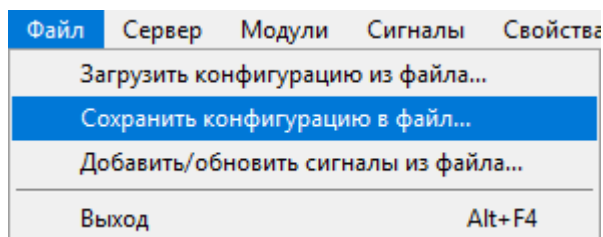
Для работы SePlatform.Imitator в режиме дополнения пропущенных исторических данных в состав конфигурации SePlatform.Data Server добавьте модуль HUB Module и выполните настройку:

1. В **Общих** параметрах модуля параметру **Активность** установите значение «Да».
2. В настройках источника в группе параметров **Настройки файлового интерфейса**:
 - В параметру **Включить интерфейс** установите значение «Да»;
 - в значении параметра **Полное имя папки для чтения данных** укажите путь к папке, из которой будут читаться файлы данных.



4.4. Сохранение конфигурации SePlatform.Data Server

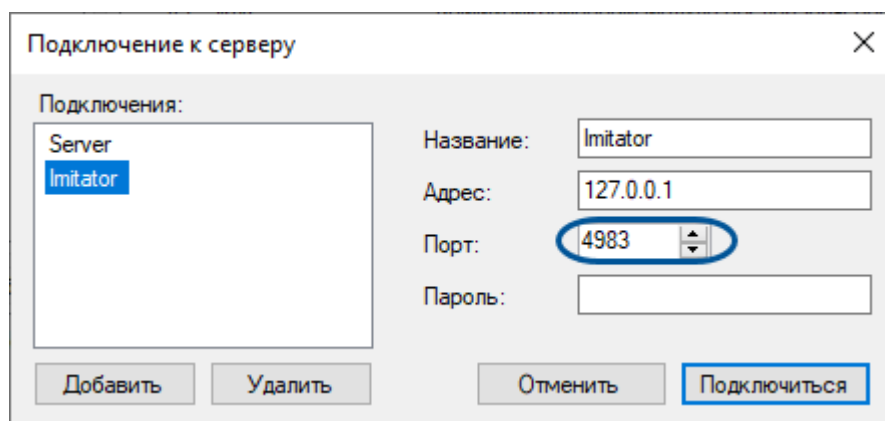
Чтобы сохранить конфигурацию настроенного SePlatform.Data Server, выполните команду меню **Файл** → **Сохранить конфигурацию в файл...** и сохраните файл конфигурации на диске.



4.5. Загрузка конфигурации в SePlatform.Imitator

Чтобы загрузить сохраненную конфигурацию SePlatform.Data Server в SePlatform.Imitator:

1. Подключитесь к SePlatform.Imitator с помощью сервисного приложения Конфигуратор. Для подключения укажите порт «4983».



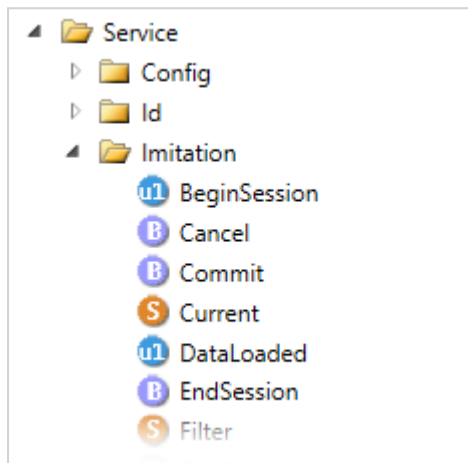
2. Выполните команду меню **Файл** → **Загрузить конфигурацию из файла...** и выберите ранее сохраненный файл конфигурации SePlatform.Data Server на диске.
3. Перезапустите службу **SePlatform.Imitator**.

5. Служебные сигналы SePlatform.Imitator

При запуске SePlatform.Imitator создаёт служебные сигналы для управления и контроля процессов воспроизведения истории, коррекции исторических данных, дополнения пропущенных исторических данных.

Полный тег служебных сигналов SePlatform.Imitator имеет вид:

```
Service.Imitation.<Имя сигнала>
```



Назначение служебных сигналов:

Тег и тип сигнала	Назначение
Управление сессией имитации	
«BeginSession» (UInt1)	Запустить новую сессию имитации в режиме: <ul style="list-style-type: none">» «0» - воспроизведение истории;» «1» - перезапись исторических данных;» «2» - дополнение пропущенных исторических данных.
«EndSession» (Bool)	Завершить сессию имитации: <ul style="list-style-type: none">» «True» - завершить сессию.
«Cancel» (Bool)	Отменить сессию имитации: <ul style="list-style-type: none">» «True» - отменить;» «False» - значение по умолчанию, устанавливается после отмены сессии имитации.
Управление имитацией	
«LoadData» (Bool)	Загрузить данные для воспроизведения: <ul style="list-style-type: none">» «true» - загрузить данные . Загрузка возможна только в рамках открытой сессии.

Тег и тип сигнала	Назначение
«SetSpeed» (Float)	<p>Установить скорость воспроизведения истории.</p> <p>Указывается время в секундах между воспроизводимыми значениями. Например, если задать значение «1», то будут отображаться данные за каждую секунду; если задать значение «10» - за каждую десятую секунду.</p> <p>Сбрасывается после установки сигналов «EndSession» и «BeginSession».</p>
«SetCurrent» (String)	<p>Установить текущее положение процесса воспроизведения истории.</p> <p>Перемещает текущую метку воспроизведения истории на заданную. Задаётся в формате <code>datetime_json</code>.</p> <p>Сбрасывается после установки сигналов «EndSession» и «BeginSession».</p>
«SetState» (Uint1)	<p>Управление процессом имитации:</p> <ul style="list-style-type: none"> ➤ «0» - пауза (только для режима воспроизведения истории); ➤ «1» - запустить процесс имитации. <p>Воспроизведение истории возможно только после полной загрузки имитационных данных.</p>
«Commit» (Bool)	<p>Применить результаты имитации:</p> <ul style="list-style-type: none"> ➤ «True» - сохранить результат в <code>SePlatform.Historian</code>; ➤ «False» - значение по умолчанию, устанавливается после завершения сохранения.
Параметры имитации	
«IntervalStart» (String)	<p>Метка времени начала интервала имитации.</p> <p>Задаётся в формате <code>datetime_json</code>. Сбрасывается после установки сигналов «EndSession» и «BeginSession».</p> <p>Значение по умолчанию: "" - левая граница отсутствует.</p>
«SetIntervalStart» (String)	<p>Задать метку времени начала интервала имитации.</p> <p>Задаётся в формате <code>datetime_json</code>.</p>
«IntervalEnd» (String)	<p>Метка времени конца интервала имитации.</p> <p>Задаётся в формате <code>datetime_json</code>. Сбрасывается после установки сигналов «EndSession» и «BeginSession».</p> <p>Значение по умолчанию: "" - правая граница отсутствует.</p>
«SetIntervalEnd» (String)	<p>Задать метку времени конца интервала имитации.</p> <p>Задаётся в формате <code>datetime_json</code>.</p>

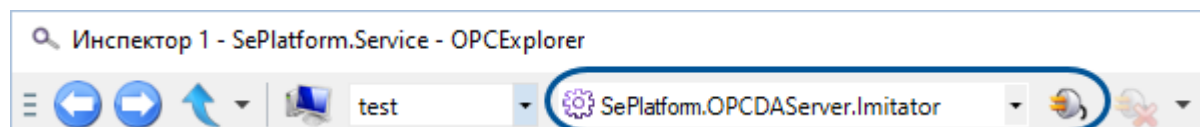
Тег и тип сигнала	Назначение
«Filter» (String)	Фильтр по тегам. Сбрасывается после установки сигналов «EndSession» и «BeginSession». Значение по умолчанию: "" - все теги.
«SetFilter» (String)	Установить фильтр по тегам.
Контроль текущего состояния сессии имитации	
«DataLoaded» (Uint1)	Процент готовности данных к воспроизведению от «0» до «100». Сбрасывается после установки сигналов «EndSession» и «BeginSession». Значение по умолчанию: «0».
«Current» (String)	Текущее положение процесса воспроизведения истории. Отображается в формате <code>datetime_json</code> . Сбрасывается после установки сигналов «EndSession» и «BeginSession». Значение по умолчанию: "".
«Speed» (Float)	Скорость воспроизведения истории. Время в секундах между воспроизводимыми значениями. Сбрасывается после установки сигналов «EndSession» и «BeginSession». Значение по умолчанию: «1».
«State» (Uint1)	Статус имитации: <ul style="list-style-type: none"> ➤ «0» - остановлено или завершено; ➤ «1» - воспроизводится; ➤ «2» - успешно завершено (для режима перезаписи и дополнения истории); ➤ «3» - завершено с ошибкой (для режима перезаписи и дополнения истории). Сбрасывается после установки сигналов «EndSession» и «BeginSession». Значение по умолчанию: «0».
«SessionInProgress» (Bool)	Флаг наличия активной сессии. Устанавливается после установки сигнала «BeginSession», сбрасывается после установки сигнала «EndSession». Значение по умолчанию: «False».
Информация об ошибках	
«LastError» (String)	Информация о последней возникшей ошибке.
Работа с данными	

Тег и тип сигнала	Назначение
«Override» (String)	<p>Задать массив перезаписываемых значений тега. Задается в формате trend_json.</p> <p>Позволяет перезаписать значения нескольких тегов. Для этого следует в значении сигнала «Override» в формате trend_json поочередно задать массивы перезаписываемых значений каждого тега.</p>
«GetOverriden» (Bool)	<p>Получить массив перезаписываемых тегов:</p> <ul style="list-style-type: none"> ➤ «True» - получить массив; ➤ «False» - значение по умолчанию.
«GetOverridenResponse» (String)	<p>Массив перезаписываемых тегов в формате:</p> <pre>["Tag": "<string>", ...]</pre>
«ReadOverriden» (String)	<p>Получить массив перезаписываемых значений:</p> <pre>{ "Tag": "<string>" }</pre>
«ReadOverridenResponse» (String)	<p>Массив перезаписываемых значений. Строка в формате trend_json со значениями за интервал, заданный в сигналах «IntervalStart» и «IntervalEnd».</p>
«ReadHistory» (String)	<p>Запросить историю значений тега:</p> <pre>{ "Tag": "<string>" }</pre>
«ReadHistoryResponse» (String)	<p>Массив исторических значений. Строка в формате trend_json со значениями за интервал, заданный в сигналах «IntervalStart» и «IntervalEnd».</p>

6. Работа с SePlatform.Imitator

Управление и контроль процессами воспроизведения истории, коррекции исторических данных, дополнения пропущенных исторических данных выполняется с помощью служебных сигналов SePlatform.Imitator.

Чтобы изменять и контролировать значения служебных сигналов, подключитесь к SePlatform.Imitator, с помощью OPC клиента, например Service - OPCExplorer. Для этого подключитесь к источнику «SePlatform.OPCDAServer.Imitator».

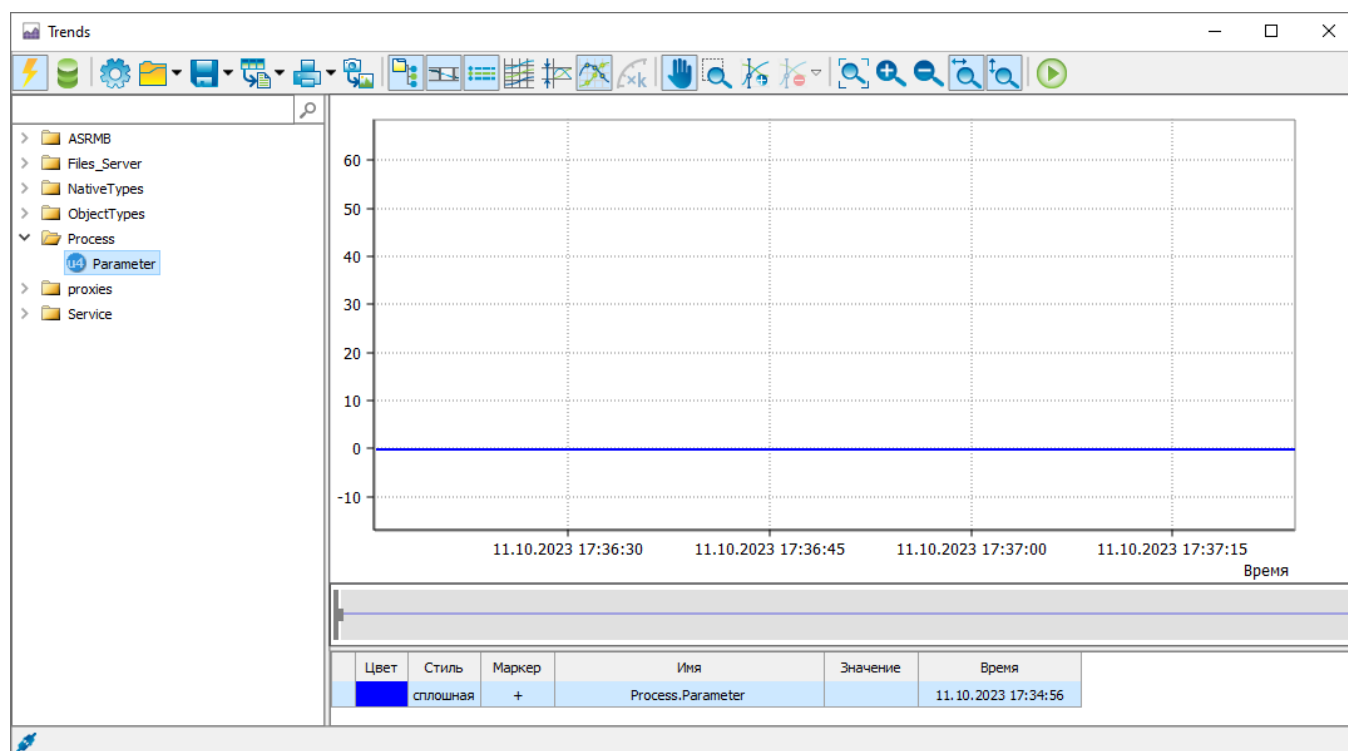


Подробная информация о работе SePlatform.Imitator выводится в журнал приложений. Для просмотра журнала запустите приложение Service - LogViewer.

6.1. Воспроизведение истории

Для просмотра воспроизведения истории с помощью SePlatform.Imitator запустите приложение SePlatform.HMI.Trends, у которого в качестве источника данных должен быть настроен SePlatform.Imitator.

В оперативном режиме на трендовое поле добавьте сигналы из дерева SePlatform.Imitator, для которых нужно воспроизвести историю изменения значений.



ОБРАТИТЕ ВНИМАНИЕ

В имитационной базе должна быть история изменений по добавленным сигналам: значения сигналов изменялись после настроек SePlatform.Historian и SePlatform.Data Server.

В Service - OPCExplorer в Инспектор добавьте служебные сигналы для управления и контроля процесса воспроизведения истории:

- «SessionInProgress» - флаг наличия активной сессии;
- «EndSession» - команда завершения сессии проигрывания;
- «BeginSession» - команда запуска сессии имитации;
- «SetIntervalStart» - задать метку времени начала имитируемого интервала;
- «SetIntervalEnd» - задать метку времени конца имитируемого интервала;
- «SetSpeed» - задать скорость проигрывания;
- «IntervalStart» - метка времени начала интервала проигрывания;
- «IntervalEnd» - метка времени конца интервала проигрывания;
- «Speed» - текущая скорость проигрывания;
- «LoadData» - загрузить данные для проигрывания;
- «DataLoaded» - процент готовности данных к проигрыванию;
- «SetState» - запустить процесс имитации;
- «Current» - текущее положение процесса проигрывания;
- «State» - статус процесса имитации.

Для воспроизведения истории:

1. Убедитесь, что нет активной сессии имитации:

- «SessionInProgress» == «False».

Если «SessionInProgress» == «True», то завершите текущую сессию имитации, установив значение сигнала:

- «EndSession» = «True».

Тип	Сигнал	Значение	Описание
B	Service.Imitation.SessionInProgress	False	Флаг наличия активной сессии
B	Service.Imitation.EndSession	True	Завершить сессию проигрывания

2. Запустите новую сессию имитации в режиме воспроизведения истории, установив значение сигнала:

- «BeginSession» = «0».

При этом активируется флаг наличия активной сессии «SessionInProgress» == «True».

Тип	Сигнал	Значение	Описание
B	Service.Imitation.SessionInProgress	True	Флаг наличия активной сессии
B	Service.Imitation.EndSession	True	Завершить сессию проигрывания
U1	Service.Imitation.BeginSession	0	Запустить сессию имитации

3. Задайте временной интервал в формате `datetime_json`, за который требуется воспроизвести историю, а также скорость проигрывания, установив значения сигналов:

- «SetIntervalStart»;
- «SetIntervalEnd»;
- «SetSpeed».

Пример метки времени в формате `datetime_json`, соответствующий времени «11.10.2023 17:32:01.1» в часовом поясе UTC+7:

```
{ "y": 2023, "mo": 10, "d": 11, "h": 10, "m": 32, "s": 1, "ms": 1 }
```

Сигналы устанавливают значения соответствующих сигналов «IntervalStart», «IntervalEnd» и «Speed».

Тип	Сигнал	Значение	Описание
S	Service.Imitation.SetIntervalStart	{"y":2023,"mo":10,"d":11,"h":10,"m":32,"s":1,"ms":1}	Задать метку времени начала имитируемого интервала
S	Service.Imitation.SetIntervalEnd	{"y":2023,"mo":10,"d":11,"h":10,"m":35,"s":1,"ms":1}	Задать метку времени конца имитируемого интервала
f4	Service.Imitation.SetSpeed	1	Установить скорость проигрывания
S	Service.Imitation.IntervalStart	{"y":2023,"mo":10,"d":11,"h":10,"m":35,"s":1,"ms":1}	Метка времени начала интервала проигрывания
S	Service.Imitation.IntervalEnd	{"y":2023,"mo":10,"d":11,"h":10,"m":32,"s":1,"ms":1}	Метка времени конца интервала проигрывания
f4	Service.Imitation.Speed	1	Скорость проигрывания

4. Загрузите данные для воспроизведения истории, установив значение сигнала:

➤ «LoadData» = «True».

Процент загруженных данных отображается в значении сигнала «DataLoaded».

Тип	Сигнал	Значение	Описание
B	Service.Imitation.LoadData	True	Загрузить данные для проигрывания
u1	Service.Imitation.DataLoaded	100	Процент готовности данных к проигрыванию

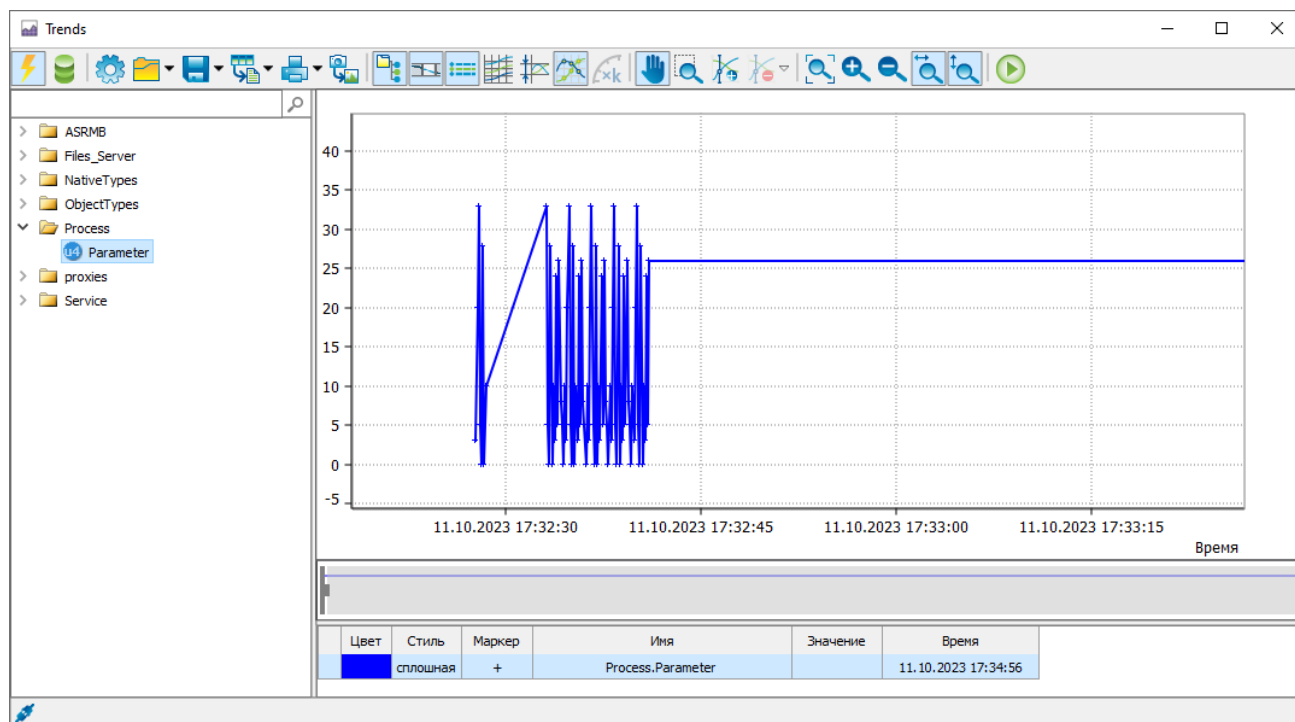
5. После полной загрузки данных («DataLoaded» == «100») запустите воспроизведение истории, установив значение сигнала:

➤ «SetState» = «1».

В процессе воспроизведения истории текущее положение процесса воспроизведения отображается в значении сигнала «Current», а статус воспроизведения «State» == «1».

Тип	Сигнал	Значение	Описание
u1	Service.Imitation.SetState	1	Управление процессом имитации
S	Service.Imitation.Current	{"y":2023,"mo":10,"d":11,"h":10,"m":32,"s":4,"ms":76}	Текущее положение процесса проигрывания
u1	Service.Imitation.State	1	Статус процесса имитации

На трендовом поле SePlatform.HMI.Trends в оперативном режиме отображается изменение значений сигналов за воспроизводимый период времени.



6. Чтобы завершить сессию воспроизведения истории, установите значение сигнала:

➤ «EndSession» = «True».

При этом сбрасываются значения сигналов «IntervalStart», «IntervalEnd», «Current», а также устанавливаются следующие значения сигналов:

➤ «State» = «0»;

➤ «Speed» = «1»;

➤ «SessionInProgress» = «False».

Тип	Сигнал	Значение	Описание
B	Service.Imitation.EndSession	True	Завершить сессию проигрывания
S	Service.Imitation.IntervalStart		Метка времени начала интервала проигрывания
S	Service.Imitation.IntervalEnd		Метка времени конца интервала проигрывания
S	Service.Imitation.Current		Текущее положение процесса проигрывания
u1	Service.Imitation.State	0	Статус процесса имитации
f4	Service.Imitation.Speed	1	Скорость проигрывания
B	Service.Imitation.SessionInProgress	False	Флаг наличия активной сессии



ПРИМЕЧАНИЕ

В процессе воспроизведения истории («State» == «1») возможно:

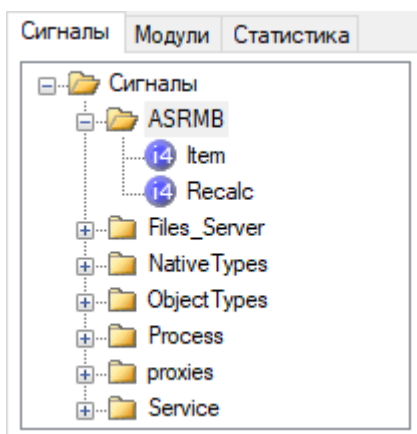
1. Изменение скорости воспроизведения в сигнале «SetSpeed».
2. Изменение текущего положения воспроизведения в сигнале «SetCurrent».
3. Приостановка воспроизведения (пауза): «SetState» = «0».
4. Завершение сессии воспроизведения: «EndSession» = «True».

При достижении конца воспроизводимого интервала или приостановке воспроизведения («State» == «0») возможно:

1. Изменение скорости воспроизведения истории в сигнале «SetSpeed».
2. Изменение текущего положения воспроизведения в сигнале «SetCurrent».
3. Возобновление воспроизведения: «SetState» = «1».
4. Завершение сессии воспроизведения: «EndSession» = «True».

6.2. Перезапись исторических данных

В SePlatform.Historian хранится некоторая история изменения сигнала «ASRMB.Recalc» и вычисленных значений сигнала «ASRMB.Item».



Значение сигнала «ASRMB.Item» вычисляется на основе значения сигнала «ASRMB.Recalc» по формуле:

```
ASRMB.Item = ASRMB.Recalc + 10
```

Требуется пересчитать и перезаписать в SePlatform.Historian значения сигнала «ASRMB.Item» на основе новых значений сигнала «ASRMB.Recalc».

6.2.1. Перезапись единственного значения

В Service - OPCExplorer в Инспектор добавьте служебные сигналы для управления и контроля процесса перезаписи исторических данных:

- «SessionInProgress» - флаг наличия активной сессии;
- «BeginSession» - команда запуска сессии имитации;
- «Cancel» - отменить сессию имитации;
- «SetIntervalStart» - задать метку времени начала перезаписываемого интервала;
- «SetIntervalEnd» - задать метку времени конца перезаписываемого интервала;
- «IntervalStart» - метка времени начала перезаписываемого интервала;
- «IntervalEnd» - метка времени конца перезаписываемого интервала;
- «ReadHistory» - запросить историю значений тега;
- «ReadHistoryResponse» - список исторических значений;
- «SetState» - запустить процесс имитации;
- «State» - статус процесса имитации;
- «Commit» - применить результаты имитации.

Для перезаписи единственного значения:

1. Переведите SePlatform.Imitator в работу, установив значение сигнала «Service.State.Server.Set» = «True».
2. Убедитесь, что нет активной сессии имитации:

- «SessionInProgress» == «False».

Если «SessionInProgress» == «True», то отмените текущую сессию имитации, установив значение сигнала:

- «Cancel» = «True».

Тип	Сигнал	Значение	Описание
B	Service.Imitation.SessionInProgress	False	Флаг наличия активной сессии
B	Service.Imitation.Cancel	True	Отменить результаты имитации

3. Запустите новую сессию имитации в режиме перезаписи исторических данных, установив значение сигнала

- «BeginSession» = «1».

Тип	Сигнал	Значение	Описание
B	Service.Imitation.SessionInProgress	False	Флаг наличия активной сессии
B	Service.Imitation.Cancel	True	Отменить результаты имитации
U1	Service.Imitation.BeginSession	1	Запустить сессию имитации

4. Задайте временной интервал, за который требуется перезаписать данные в SePlatform.Historian, установив значения сигналов:

- «SetIntervalStart»;
- «SetIntervalEnd».

Значения в указанных сигналах задаются в формате `datetime_json` по времени UTC. Например, требуется перезаписать данные в интервале с «12.10.2023 10:38:39.1» по «12.10.2023 10:38:40.1»:

➤ в значение сигнала «`SetIntervalStart`» запишите строку:

```
{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 38, "s": 39, "ms": 1 }
```

➤ в значение сигнала «`SetIntervalEnd`» запишите строку:

```
{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 38, "s": 40, "ms": 1 }
```

Сигналы устанавливают значения соответствующих сигналов «`IntervalStart`» и «`IntervalEnd`».

Тип	Сигнал	Значение	Описание
S	Service.Imitation.IntervalStart	{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 38, "s": 40, "ms": 1 }	Метка времени начала интервала проигрывания
S	Service.Imitation.IntervalEnd	{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 38, "s": 39, "ms": 1 }	Метка времени конца интервала проигрывания
S	Service.Imitation.SetIntervalStart	{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 38, "s": 39, "ms": 1 }	Задать метку времени начала имитируемого интервала
S	Service.Imitation.SetIntervalEnd	{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 38, "s": 40, "ms": 1 }	Задать метку времени конца имитируемого интервала

5. Проверьте значения тегов в `SePlatform.Historian` за заданный временной интервал. Для этого в сигнале «`ReadHistory`» укажите тег в формате:

```
{ "Tag": "ASRMB.Recalc" }
```

В значении сигнала «`ReadHistoryResponse`» отобразятся значения тега за указанный интервал.

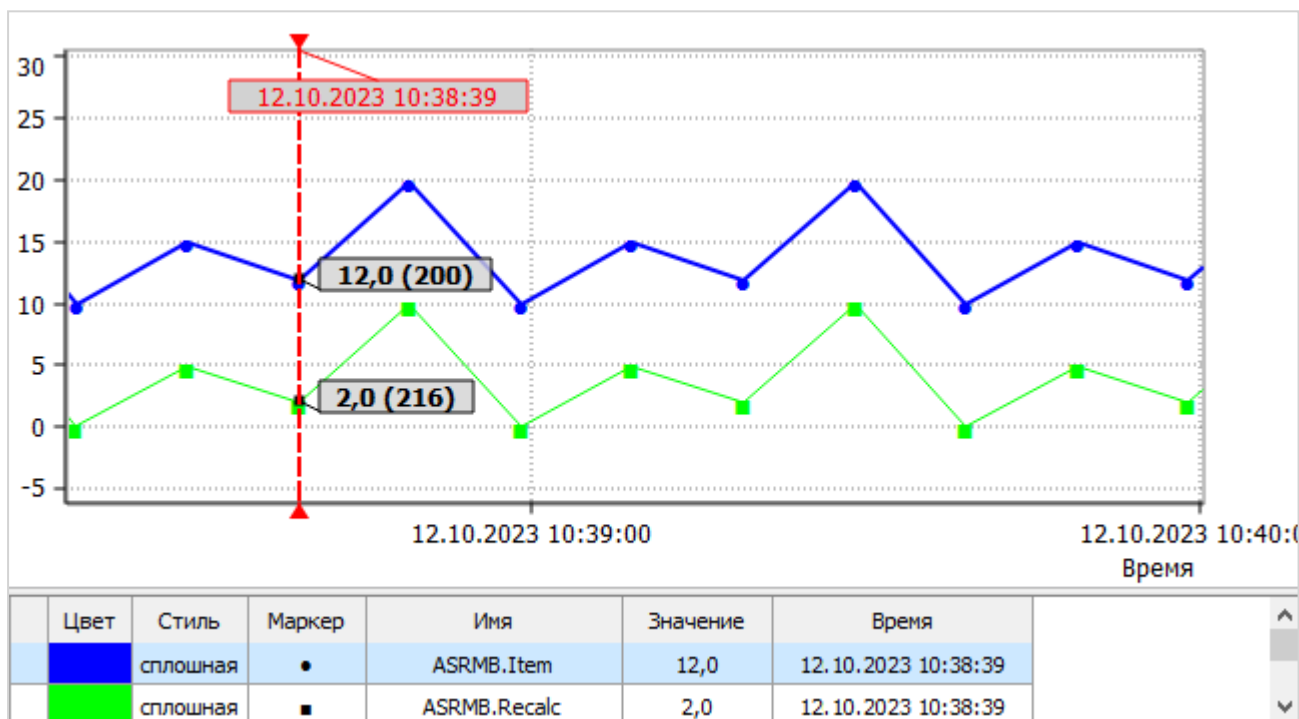
История значений тега «`ASRMB.Recalc`» в указанном интервале:

Тип	Сигнал	Значение	Описание
S	Service.Imitation.ReadHistory	{ "Tag": "ASRMB.Recalc" }	Запросить список исторических значений
S	Service.Imitation.ReadHistoryResponse	<pre>{ "Tag": "ASRMB.Recalc", "Values": [{ "v": "2", "q": "216", "t": { "y": "2023", "mo": "10", "d": "12", "h": "3", "m": "38", "s": "39", "ms": "12" } }] }</pre>	Список исторических значений

История значений тега «`ASRMB.Item`» в указанном интервале:

Тип	Сигнал	Значение	Описание
S	Service.Imitation.ReadHistory	{"Tag": "ASRMB.Item"}	Запросить список исторических значений
S	Service.Imitation.ReadHistoryResponse	<pre> { "Tag": "ASRMB.Item", "Values": [{ "v": "12", "q": "216", "t": { "y": "2023", "mo": "10", "d": "12", "h": "3", "m": "38", "s": "39", "ms": "12" } }] }</pre>	Список исторических значений

Графики сигналов имеют вид:



6. Установите новое значение тегу «ASRMB.Recalc», на основе которого требуется пересчитать и перезаписать значение тега «ASRMB.Item». Например, «ASRMB.Recalc» == «15».

Тип	Сигнал	Значение	Описание
i4	ASRMB.Recalc	15	
i4	ASRMB.Item		

7. Запустите имитацию, установив значение сигнала:

➤ «SetState» = «1».

В результате имитации значение тега «ASRMB.Item» будет пересчитано, а статус процесса имитации примет значение «State» == «2».

Тип	Сигнал	Значение	Описание
u1	Service.Imitation.State	2	Статус процесса имитации
u1	Service.Imitation.SetState	1	Управление процессом имитации
i4	ASRMB.Recalc	15	
i4	ASRMB.Item	25	

8. Сохраните результат имитации в SePlatform.Historian, установив значение сигнала:

➤ «Commit» = «True».

Новые значения тегов «ASRMB.Recalc» и «ASRMB.Item» будут перезаписаны в SePlatform.Historian. При этом сбрасываются значения сигналов «IntervalStart» и «IntervalEnd», а также устанавливаются следующие значения сигналов:

➤ «State» = «0»;

➤ «SessionInProgress» = «False».

Тип	Сигнал	Значение	Описание
B	Service.Imitation.SessionInProgress	False	Флаг наличия активной сессии
S	Service.Imitation.IntervalStart		Метка времени начала интервала проигрывания
S	Service.Imitation.IntervalEnd		Метка времени конца интервала проигрывания
UI	Service.Imitation.State	0	Статус процесса имитации
B	Service.Imitation.Commit	True	Применить результаты имитации

Чтобы проверить запись новых значений тегов в SePlatform.Historian за заданный временной интервал:

1. Запустите сессию имитации: «BeginSession» = «1».

2. Задайте временной интервал:

➤ в значение сигнала «SetIntervalStart» запишите строку:

```
{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 38, "s": 39, "ms": 1 }
```

➤ в значение сигнала «SetIntervalEnd» запишите строку:

```
{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 38, "s": 40, "ms": 1 }
```

3. В сигнале «ReadHistory» укажите тег, после чего сигнале «ReadHistoryResponse» отобразятся значения тега за указанный интервал.

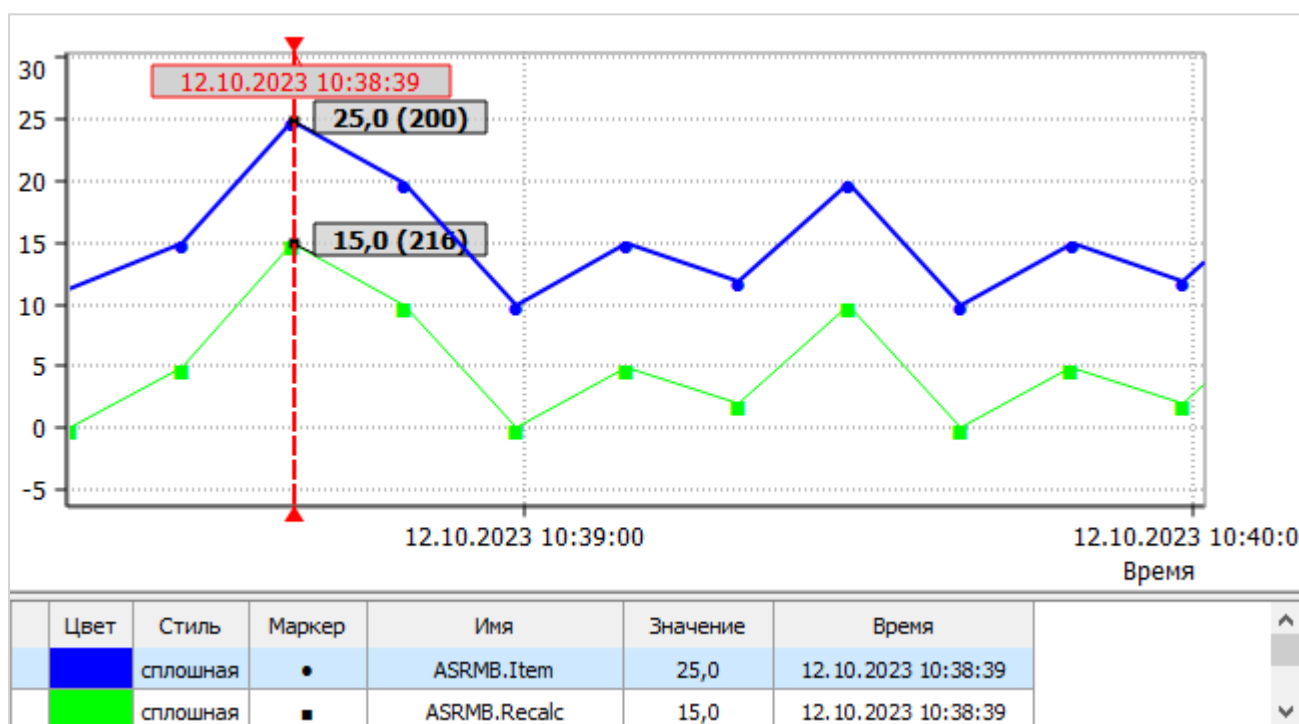
История значений тега «ASRMB.Recalc» в указанном интервале:

Тип	Сигнал	Значение	Описание
S	Service.Imitation.ReadHistory	{ "Tag": "ASRMB.Recalc" }	Запросить список исторических значений
S	Service.Imitation.ReadHistoryResponse	<pre>{ "Tag": "ASRMB.Recalc", "Values": [{ "v": "15", "q": "216", "t": { "y": "2023", "mo": "10", "d": "12", "h": "3", "m": "38", "s": "39", "ms": "1" } }] }</pre>	Список исторических значений

История значений тега «ASRMB.Item» в указанном интервале:

Тип	Сигнал	Значение	Описание
S	Service.Imitation.ReadHistory	<code>{"Tag": "ASRMB.Item"}</code>	Запросить список исторических значений
S	Service.Imitation.ReadHistoryResponse	<pre> { "Tag": "ASRMB.Item", "Values": [{ "v": "25", "q": "200", "t": { "y": "2023", "mo": "10", "d": "12", "h": "3", "m": "38", "s": "39", "ms": "12" } }] }</pre>	Список исторических значений

Графики сигналов с перезаписанными значениями примут вид:



6.2.2. Перезапись массива данных

В Service - OPCExplorer в Инспектор добавьте служебные сигналы для управления и контроля процесса перезаписи исторических данных:

- «SessionInProgress» - флаг наличия активной сессии;
- «BeginSession» - команда запуска сессии имитации;
- «Cancel» - отменить сессию имитации;
- «SetIntervalStart» - задать метку времени начала перезаписываемого интервала;
- «SetIntervalEnd» - задать метку времени конца перезаписываемого интервала;
- «IntervalStart» - метка времени начала перезаписываемого интервала;
- «IntervalEnd» - метка времени конца перезаписываемого интервала;

- «ReadHistory» - запросить список исторических значений;
- «ReadHistoryResponse» - список исторических значений;
- «Override» - задать перезаписываемые сигналы;
- «ReadOverriden» - запросить список переопределяемых значений;
- «ReadOverridenResponse» - список переопределяемых значений;
- «GetOverriden» - запросить список переопределяемых тегов;
- «GetOverridenResponse» - список переопределяемых тегов;
- «SetState» - запустить процесс имитации;
- «State» - статус процесса имитации;
- «Commit» - применить результаты имитации.

Для перезаписи массива значений:

1. Переведите SePlatform.Imitator в работу, установив значение сигнала «Service.State.Server.Set» = «True».
2. Убедитесь, что нет активной сессии имитации:

- «SessionInProgress» == «False».

Если «SessionInProgress» == «True», то отмените текущую сессию имитации, установив значение сигнала:

- «Cancel» = «True».

Тип	Сигнал	Значение	Описание
B	Service.Imitation.SessionInProgress	False	Флаг наличия активной сессии
B	Service.Imitation.Cancel	True	Отменить результаты имитации

3. Запустите новую сессию имитации в режиме перезаписи исторических данных, установив значение сигнала

- «BeginSession» = «1».

Тип	Сигнал	Значение	Описание
B	Service.Imitation.SessionInProgress	False	Флаг наличия активной сессии
B	Service.Imitation.Cancel	True	Отменить результаты имитации
U1	Service.Imitation.BeginSession	1	Запустить сессию имитации

4. Задайте временной интервал, за который требуется перезаписать данные в SePlatform.Historian, установив значения сигналов:

➤ «SetIntervalStart»;

➤ «SetIntervalEnd».

Значения в указанных сигналах задаются в формате `datetime_json` по времени UTC. Например, требуется перезаписать данные в интервале с «12.10.2023 10:56:19.1» по «12.10.2023 10:56:30.1»:

➤ в значение сигнала «SetIntervalStart» запишите строку:

```
{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 56, "s": 19, "ms": 1 }
```

➤ в значение сигнала «SetIntervalEnd» запишите строку:

```
{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 56, "s": 30, "ms": 1 }
```

Сигналы устанавливают значения соответствующих сигналов «IntervalStart» и «IntervalEnd».

Тип	Сигнал	Значение	Описание
S	Service.Imitation.SetIntervalStart	{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 56, "s": 19, "ms": 1 }	Задать метку времени начала имитируемого интер...
S	Service.Imitation.SetIntervalEnd	{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 56, "s": 30, "ms": 1 }	Задать метку времени конца имитируемого интер...
S	Service.Imitation.IntervalEnd	{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 56, "s": 30, "ms": 1 }	Метка времени конца интервала проигрывания
S	Service.Imitation.IntervalStart	{ "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 56, "s": 19, "ms": 1 }	Метка времени начала интервала проигрывания

5. Проверьте значения тегов в SePlatform.Historian за заданный временной интервал. Для этого в сигнале «ReadHistory» укажите тег в формате:

```
{ "Tag": "ASRMB.Recalc" }
```

В значении сигнала «ReadHistoryResponse» отобразятся значения тега за указанный интервал.

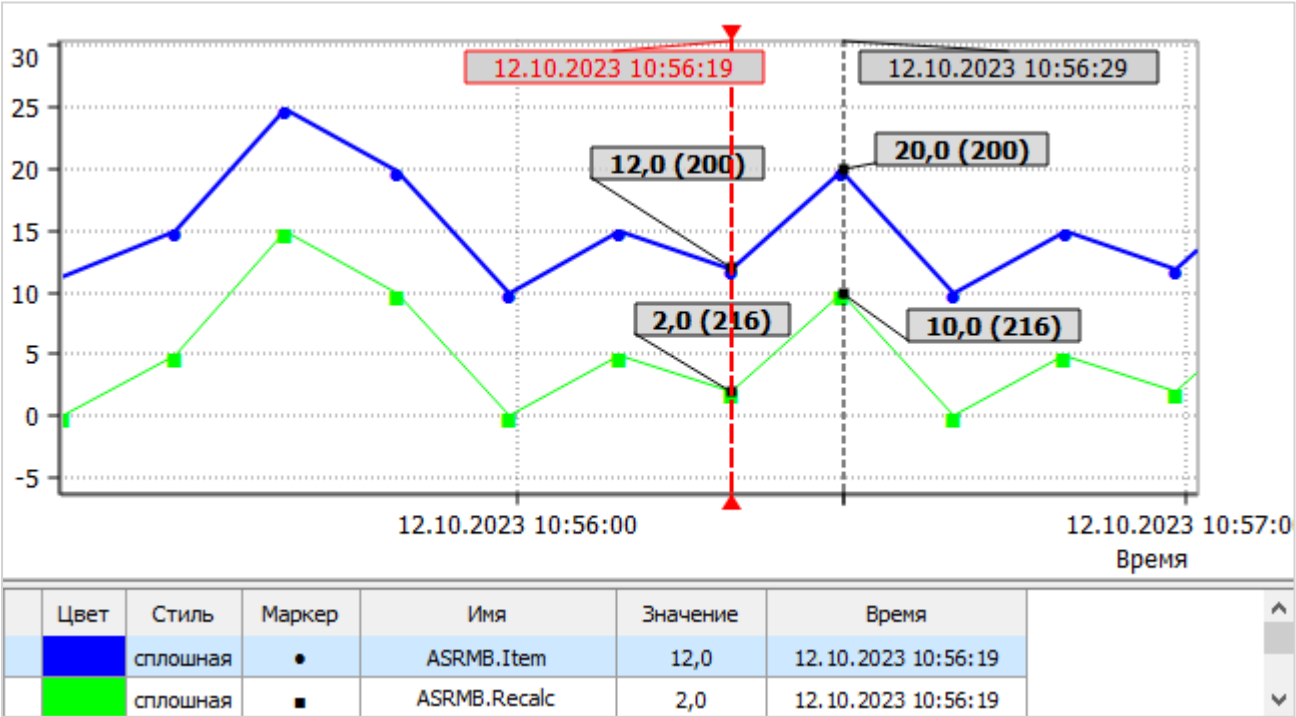
История значений тега «ASRMB.Recalc» в указанном интервале содержит две записи:

Тип	Сигнал	Значение	Описание
S	Service.Imitation.ReadHistory	{ "Tag": "ASRMB.Recalc" }	Запросить список исторических значений
S	Service.Imitation.ReadHistoryResponse	<pre>{ "Tag": "ASRMB.Recalc", "Values": [{ "v": "2", "q": "216", "t": { "y": "2023", "mo": "10", "d": "12", "h": "3", "m": "56", "s": "19", "ms": "1" } }] }</pre>	Список исторических значений

История значений тега «ASRMB.Item» в указанном интервале содержит две записи:

Тип	Сигнал	Значение	Описание
S	Service.Imitation.ReadHistory	{ "Tag": "ASRMB.Item" }	Запросить список исторических значений
<pre>{ "Tag": "ASRMB.Item", "Values": [{ "v": "12", "q": "200", "t": { "y": "2023", "mo": "10", "d": "12", "h": "3", "m": "56", "s": "19", "ms": "1" } }] }</pre>			
S	Service.Imitation.ReadHistoryResponse	{ "v": "20", "q": "200", "t": { "y": "2023", "mo": "10", "d": "12", "h": "3", "m": "56", "s": "29", "ms": "1" } } }	Список исторических значений

Графики сигналов имеют вид:



6. Установите новые значения тега «ASRMB.Recalc», на основе которых требуется пересчитать и перезаписать значения тега «ASRMB.Item». Например, тегу «ASRMB.Recalc» установите значения «15» и «0», вместо имеющихся «2» и «10». Для этого в сигнал «Override» запишите команду на перезапись массива значений в формате `trend_json`:

```
{
  "Tag": "ASRMB.Recalc",
  "Add": [
    {
      "v": "15",
      "q": "216",
      "t": {
        "y": 2023,
        "mo": 10,
        "d": 12,
        "h": 3,
        "m": 56,
        "s": 19,
        "ms": 1
      }
    },
    {
      "v": "0",
      "q": "216",
      "t": {
        "y": 2023,
        "mo": 10,
        "d": 12,
        "h": 3,
        "m": 56,
        "s": 29,
        "ms": 1
      }
    }
  ]
}
```

7. Убедитесь, что для перезаписи подготовлены верные значения - «15» и «0». Для этого укажите в сигнале «ReadOverriden» тег, значения которого требуется перезаписать, в формате:

```
{
  "Tag": "ASRMB.Recalc"
}
```

Массив значений, подготовленный для перезаписи, будет выведен в значении сигнала «ReadOverridenResponse».

Тип	Сигнал	Значение	Описание
S	Service.Imitation.Override	{ "Tag": "ASRMB.Recalc", "Add": [{ "v": "15", "q": "216", "t": { "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 56, "s": 19, "ms": 1 } }, { "v": "0", "q": "216", "t": { "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 56, "s": 29, "ms": 1 } }] }	Задать перезаписываемые сигналы
S	Service.Imitation.ReadOverriden	{ "Tag": "ASRMB.Recalc" }	Запросить список переопределяемых значений
S	Service.Imitation.ReadOverridenResponse	{ "Tag": "ASRMB.Recalc", "Values": [{ "v": "15", "q": "216", "t": { "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 56, "s": 19, "ms": 1 } }, { "v": "0", "q": "216", "t": { "y": 2023, "mo": 10, "d": 12, "h": 3, "m": 56, "s": 29, "ms": 1 } }] }	Список переопределяемых значений

8. Убедитесь, что список перезаписываемых тегов верен, установив значение сигнала «GetOverriden» = «True». Список перезаписываемых тегов будет выведен в значении сигнала «GetOverridenResponse».

Тип	Сигнал	Значение	Описание
B	Service.Imitation.GetOverriden	True	Запросить список переопределяемых тегов
S	Service.Imitation.GetOverridenResponse	{ "Tags": [{ "Tag": "ASRMB.Recalc" }] }	Список переопределяемых тегов

9. Запустите имитацию, установив значение сигнала:

- «SetState» = «1».

В результате имитации заданные значения тега «ASRMB.Recalc» и значения тега «ASRMB.Item» будут пересчитаны и перезаписаны. В сигналах будут отображаться последние перезаписанные значения из массива, а статус процесса имитации примет значение «State» == «2».

Тип	Сигнал	Значение	Описание
u1	Service.Imitation.State	2	Статус процесса имитации
u1	Service.Imitation.SetState	1	Управление процессом имитации
i4	ASRMB.Recalc	0	
i4	ASRMB.Item	10	

10. Сохраните результат имитации в SePlatform.Historian, установив значение сигнала: «Commit» = «True».

Массивы новых значений тегов «ASRMB.Recalc» и «ASRMB.Item» будут перезаписаны в SePlatform.Historian. При этом сбрасываются значения сигналов «IntervalStart», «IntervalEnd», «Current», а также устанавливаются следующие значения сигналов:

- «State» = «0»;
- «SessionInProgress» = «False».

Тип	Сигнал	Значение	Описание
B	Service.Imitation.SessionInProgress	False	Флаг наличия активной сессии
S	Service.Imitation.IntervalStart		Метка времени начала интервала проигрывания
S	Service.Imitation.IntervalEnd		Метка времени конца интервала проигрывания
u1	Service.Imitation.State	0	Статус процесса имитации
B	Service.Imitation.Commit	True	Применить результаты имитации

Чтобы проверить запись массивов новых значений тегов в SePlatform.Historian за заданный временной интервал:

1. Запустите сессию имитации: «BeginSession» = «1».
2. Задайте временной интервал:
 - в значение сигнала «SetIntervalStart» запишите строку:

```
{"y":2023,"mo":10,"d":12,"h":3,"m":38,"s":39,"ms":1}
```

- в значение сигнала «SetIntervalEnd» запишите строку:

```
{"y":2023,"mo":10,"d":12,"h":3,"m":38,"s":40,"ms":1}
```

3. В сигнале «ReadHistory» укажите тег, после чего в сигнале «ReadHistoryResponse» отобразятся значения тега за указанный интервал.

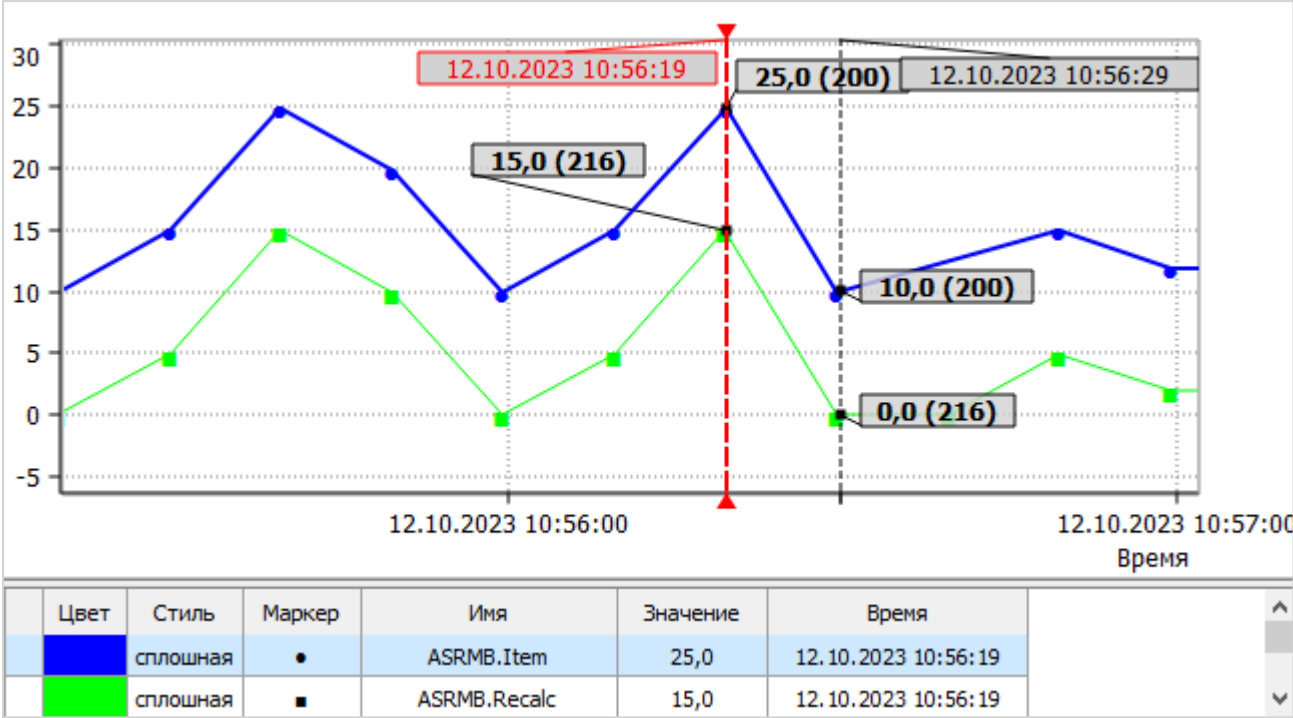
История значений тега «ASRMB.Recalc» в указанном интервале:

Тип	Сигнал	Значение	Описание
S	Service.Imitation.ReadHistory	{"Tag":"ASRMB.Recalc"}	Запросить список исторических значений
		{ "Tag": "ASRMB.Recalc", "Values": [{ "v": "15", "q": "216", "t": { "y": "2023", "mo": "10", "d": "12", "h": "3", "m": "56", "s": "19", "ms": "1" } }] }	
S	Service.Imitation.ReadHistoryResponse	{ "v": "0", "q": "216", "t": { "y": "2023", "mo": "10", "d": "12", "h": "3", "m": "56", "s": "29", "ms": "1" } }	Список исторических значений

История значений тега «ASRMB.Item» в указанном интервале:

Тип	Сигнал	Значение	Описание
S	Service.Imitation.ReadHistory	{"Tag":"ASRMB.Item"}	Запросить список исторических значений
		{ "Tag": "ASRMB.Item", "Values": [{ "v": "25", "q": "216", "t": { "y": "2023", "mo": "10", "d": "12", "h": "3", "m": "56", "s": "19", "ms": "1" } }] }	
S	Service.Imitation.ReadHistoryResponse	{ "v": "10", "q": "216", "t": { "y": "2023", "mo": "10", "d": "12", "h": "3", "m": "56", "s": "29", "ms": "1" } }	Список исторических значений

Графики сигналов с перезаписанными значениями примут вид:



6.3. Дополнение пропущенных исторических данных

Чтобы записать отсутствующие данные в SePlatform.Historian:

- 1. Определите промежуток времени, за который отсутствуют данные в SePlatform.Historian.
- 2. Остановите SePlatform.Data Server.
- 3. В папку, указанную в настройках источника HUB Module в SePlatform.Imitator, поместите файлы данных, которые не были переданы от нижестоящих систем за интересующий промежуток времени (в примере настройки папка C:\Backfilling).
- 4. Переведите SePlatform.Imitator в работу, установив значение сигнала «Service.State.Server.Set» = «True».
- 5. Откройте новую сессию имитации в режиме дополнения пропущенных исторических данных, установив значение сигнала «BeginSession» = «2».

Тип	Сигнал	Значение	Описание
01	Service.Imitation.BeginSession	2	Запустить сессию имитации

При открытии сессии имитации в режиме дополнения пропущенных исторических данных SePlatform.Imitator начинает разбор файлов данных и записывает в SePlatform.Historian полученные значения и пересчитанные зависимые параметры, заполняя промежутки времени, за которые отсутствовали данные в истории. Прочитанные файлы удаляются из папки автоматически.

7. Приложения

Приложение А: Создание имитационной базы данных

Для воспроизведения истории с помощью SePlatform.Imitator необходимо настроить сохранение данных SePlatform.Data Server в отдельную базу SePlatform.Historian, из которой SePlatform.Imitator будет получать данные для воспроизведения истории. Такая база данных называется имитационной.

Чтобы создать имитационную базу данных в SePlatform.Historian:

1. Откройте файл конфигурации SePlatform.Historian.Server.xml (по умолчанию расположен в папке C:\Program Files\SePlatform\SePlatform.Historian).
2. Добавьте имитационную базу данных, например, «**imitdb1**».

```
<?xml version="1.0" encoding="utf-8"?>
<SePlatform.Historian.Server StatPort="3388"
DefaultPrimaryDir="C:\SePlatform.Historian\Databases">
  <tcp-server default-port=4949 idle-sessions-count=1 idle-sessions-
timeout=15>
    <server-endpoint host="0.0.0.0" />
  </tcp-server>
  <Bases>
    <Base Alias="default" />
    <Base Alias="imitdb1" />
  </Bases>
</SePlatform.Historian.Server>
```

3. Сохраните изменения в файле SePlatform.Historian.Server.xml.
4. Перезапустите службу SePlatform.Historian.Server.

Приложение В: Формат `datetime_json`

Метка времени задаётся и отображается по времени UTC в служебных сигналах в формате `datetime_json` и имеет вид:

```
{"y":<uint>,"mo":<uint>,"d":<uint>,"h":<uint>,"m":<uint>,"s":<uint>,"ms":<uint>}
```

где:

- **y** - год;
- **mo** - месяц;
- **d** - день;
- **h** - часы;
- **m** - минуты;
- **s** - секунды;
- **ms** - миллисекунды.



ОБРАТИТЕ ВНИМАНИЕ

Заполнение всех полей необязательно, при этом для пропущенных полей метки времени применяется текущая дата и время «00:00:00.000».



ПРИМЕР

Метка времени:

```
{"y":2023,"mo":10,"d":9,"h":10,"m":49,"s":15,"ms":2}
```

соответствует «**9.10.2023 17:49:15.2**» в часовом поясе UTC+7.

Приложение С: Формат trend_json

Массив данных для перезаписи задаётся в служебных сигналах в формате **trend_json** и имеет вид:

```
{
  "Tag": "<string>",
  "Add": [{"v": "<string>", "q": "<uint2>", "t": {datetime_json}}, {...}, ...],
  "Update": [{"v": "<string>", "q": "<uint2>", "t": {datetime_json}}, {...}, ...],
  "Remove": [{"v": "<string>", "q": "<uint2>", "t": {datetime_json}}, {...}, ...],
  "RemoveRange": {"begin": {datetime_json}, "end": {datetime_json}},
  "RemoveFromOverriden": true,
  "Values": [{"v": "<string>", "q": "<uint2>", "t": {datetime_json}}, {...}, ...]
}
```

где **Tag** - тег сигнала, для которого задается массив значений.

Остальные параметры могут использоваться в следующих комбинациях:

1. Установка тренда (все параметры - необязательные):

- **Add** - массив добавляемых точек тренда;
- **Update** - массив переопределяемых точек тренда;
- **Remove** - массив удаляемых точек тренда;
- **RemoveRange** - удаление точек тренда на указанном интервале.



ОБРАТИТЕ ВНИМАНИЕ

Команды **Add**, **Update**, **Remove** и **RemoveRange** применяются для формирования нового набора данных, который в ходе имитации будет записан в **SePlatform.Historian**.

2. Возврат тренда:

- **Values** - массив возвращаемых точек тренда при запросе.

3. Исключить тег из списка перезаписываемых:

- **RemoveFromOverriden**.



ПРИМЕЧАНИЕ

Точка тренда - значение (V), качество (Q) и метка времени (T - задается в формате **datetime_json**).

Пример массива данных для перезаписи:

```
{ "Tag": "ASRMB.Item.Recalc", "Add": [{"v": "100", "q": "216", "t": {"y": 2023, "mo": 10, "d": 9, "h": 10, "m": 3, "s": 23, "ms": 452}}, {"v": "190", "q": "216", "t": {"y": 2023, "mo": 10, "d": 9, "h": 10, "m": 3, "s": 37, "ms": 168}}, {"v": "300", "q": "216", "t": {"y": 2023, "mo": 10, "d": 9, "h": 10, "m": 3, "s": 30, "ms": 384}}]}
```

что соответствует записи в тег «**ASRMB.Item.Recalc**» массива данных (время по UTC):

- значение «**100**», качество «**216**», метка времени «**9.10.2023 10:03:23.452**»;
- значение «**300**», качество «**216**», метка времени «**9.10.2023 10:03:30.384**»;
- значение «**190**», качество «**216**», метка времени «**9.10.2023 10:03:37.168**».